

# Projet Darts

version 0.2

BTS SNIR LaSalle Avignon 2020

## Table des matières

### 1 Le projet

#### 1.1 Table des matières

- [README](#)
- [Changelog](#)
- [A propos](#)
- [Licence GPL](#)

#### 1.2 Informations

##### Auteur

Erwan Menella [menella.erwan@gmail.com](mailto:menella.erwan@gmail.com)

##### Date

2020

##### Version

0.2

##### Voir également

<https://svn.riouxsvn.com/darts-2020/>

### 2 Changelog

r1 | www-data | 2020-02-01 15 :03 :29 +0100 (sam. 01 févr. 2020) | 1 ligne

Creating initial repository structure

### 3 README

#### 3.1 Projet

##### 3.1.1 Présentation

Le système **DARTS** est un système numérique permettant de jouer au jeu de fléchettes électroniques.

Le système DARTS est décomposé en trois modules, dont deux modules sont réalisés par des étudiants IR :

- Module de gestion de partie (**Mobile-DARTS**) : les joueurs paramètrent et lancent la partie à partir d'une application sur un terminal mobile (sous Android) ;
- Module de détection des impacts (Cible-DARTS) : la cible est équipée de capteurs permettant d'identifier la zone impactée par les fléchettes envoyées par les joueurs ;
- Module de visualisation de partie (**Écran-DARTS**) : les joueurs, les arbitres et le public peuvent visualiser en "temps réel" le déroulement de la partie (nombre de manche, point restant dans la manche, moyenne des volées, ...) sur un écran de télévision.

### 3.1.2 Module de gestion de partie (Mobile-DARTS)

Sur le terminal mobile Android, l'application doit permettre de paramétrer et démarrer une partie.

Pour cela, les joueurs pourront :

- saisir leur nom
- paramétrer la partie :
  - le type de jeu : 501 double out, 301 double out, ...
  - le nombre de joueurs,
  - le nombre de manches gagnantes,
- lancer la partie
- gérer et visualiser le déroulement de la partie :
  - changer automatiquement de joueur
  - affichage du nombre de points
  - renseigner le module Cible d'une fléchette hors cible

Pour démarrer une partie, il est nécessaire d'être connecté en Bluetooth avec le module Cible-DARTS. La connexion avec le module Écran-DARTS n'est pas indispensable pour le fonctionnement du système DARTS. On pourra visualiser sur l'interface l'état de connexion de chaque module.

Le terminal mobile possède une base de données permettant de stocker les statistiques des joueurs pour chaque partie.

### 3.1.3 Informations

#### Auteur

Erwan Menella [menella.erwan@gmail.com](mailto:menella.erwan@gmail.com)

#### Date

2020

#### Version

0.2

#### Voir également

<https://svn.riouxsvn.com/darts-2020/>

## 4 A propos

#### Auteur

Erwan Menella [menella.erwan@gmail.com](mailto:menella.erwan@gmail.com)

#### Date

2020

#### Version

0.2

#### Voir également

<https://svn.riouxsvn.com/darts-2020/>

## 5 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

## 6 Documentation des espaces de nommage

### 6.1 Paquetage projet

#### Paquetages

- package [lasalle84](#)

### 6.2 Paquetage projet.lasalle84

#### Paquetages

- package [darts](#)

### 6.3 Paquetage projet.lasalle84.darts

#### Classes

- class [ActiviteAjouterJoueur](#)  
*Déclaration de la classe [ActiviteAjouterJoueur](#).*
- class [ActiviteCreerPartie](#)
- class [ActiviteFinPartie](#)
- class [ActivitePartie](#)  
*Déclaration de la classe [ActivitePartie](#).*
- class [ExampleInstrumentedTest](#)
- class [ExampleUnitTest](#)
- class [Joueur](#)  
*Déclaration de la classe [Joueur](#).*
- class [MainActivity](#)  
*Déclaration de la classe [MainActivity](#).*
- class [Partie](#)  
*Déclaration de la classe [Partie](#).*
- class [Peripherique](#)  
*Déclaration de la classe [Peripherique](#).*
- class [TypeJeu](#)  
*Déclaration de la classe [TypeJeu](#).*

## 7 Documentation des classes

### 7.1 Référence de la classe projet.lasalle84.darts.ActiviteAjouterJoueur

Déclaration de la classe [ActiviteAjouterJoueur](#).

Graphe de collaboration de projet.lasalle84.darts.ActiviteAjouterJoueur :

projet.lasalle84.darts.ActiviteAjouterJoueur
- boutonValider - saisir - TAG
+ finish() + initiliserWidget() + onClick() + recupererWidget() # onCreate()

#### Fonctions membres publiques

- void [finish](#) ()  
*Méthode appelée lors l'utilisateur valide.*
- void [initiliserWidget](#) ()
- void [onClick](#) (View element)  
*Méthode appelée lors d'un clic sur un widget de l'activité*
- void [recupererWidget](#) ()

#### Fonctions membres protégées

- void [onCreate](#) (Bundle savedInstanceState)  
*Méthode appelée à la création de l'activité*

#### Attributs privés

- Button [boutonValider](#)  
*Bouton valider.*
- EditText [saisir](#)  
*Zone de saisir du nom du joueur.*

#### Attributs privés statiques

- static final String [TAG](#) = "IHMAjouterJoueur"  
*le TAG de la classe pour les logs*

### 7.1.1 Description détaillée

Déclaration de la classe [ActiviteAjouterJoueur](#).

Définition à la ligne 23 du fichier [ActiviteAjouterJoueur.java](#).

### 7.1.2 Documentation des fonctions membres

#### 7.1.2.1 finish()

```
projet.lasalle84.darts.ActiviteAjouterJoueur.finish ( )
```

Méthode appelée lors l'utilisateur valide.

Définition à la ligne 94 du fichier [ActiviteAjouterJoueur.java](#).

Référencé par [projet.lasalle84.darts.ActiviteAjouterJoueur.onClick\(\)](#).

```
00094         {
00095     Log.d(TAG, "finish() ");
00096
00097     if (saisir.getText() != null)
00098     {
00099         Intent retour = new Intent();
00100         retour.putExtra("Joueur", saisir.getText().toString().trim());
00101         setResult(RESULT_OK, retour);
00102         super.finish();
00103     }
00104 }
```

#### 7.1.2.2 initiliserWidget()

```
void projet.lasalle84.darts.ActiviteAjouterJoueur.initiliserWidget ( )
```

Définition à la ligne 62 du fichier [ActiviteAjouterJoueur.java](#).

Référencé par [projet.lasalle84.darts.ActiviteAjouterJoueur.onCreate\(\)](#).

```
00063     {
00064         Log.d(TAG, "initiliserWidget () ");
00065         boutonValider.setOnClickListener(this);
00066         saisir.setOnClickListener(this);
00067     }
00068 }
```

#### 7.1.2.3 onClick()

```
projet.lasalle84.darts.ActiviteAjouterJoueur.onClick (
    View element )
```

Méthode appelée lors d'un clic sur un widget de l'activité

### Paramètres

<i>element</i>	le widget cliqué
----------------	------------------

Définition à la ligne 77 du fichier [ActiviteAjouterJoueur.java](#).

Références [projet.lasalle84.darts.ActiviteAjouterJoueur.finish\(\)](#).

```

00078     {
00079         Log.d(TAG, "OnClick()");
00080
00081         if(element == boutonValider)
00082         {
00083             Log.d(TAG, "clic BoutonValider ");
00084             finish();
00085         }
00086     }

```

#### 7.1.2.4 onCreate()

```

projet.lasalle84.darts.ActiviteAjouterJoueur.onCreate (
    Bundle savedInstanceState ) [protected]

```

Méthode appelée à la création de l'activité

### Paramètres

<i>savedInstanceState</i>	
---------------------------	--

Définition à la ligne 35 du fichier [ActiviteAjouterJoueur.java](#).

Références [projet.lasalle84.darts.ActiviteAjouterJoueur.initiliserWidget\(\)](#), et [projet.lasalle84.darts.ActiviteAjouterJoueur.recupererWidget\(\)](#).

```

00035     {
00036         Log.d(TAG, "onCreate()");
00037         super.onCreate(savedInstanceState);
00038         setContentView(R.layout.activity_ajouter_joueur);
00039         recupererWidget();
00040         initiliserWidget();
00041     }

```

#### 7.1.2.5 recupererWidget()

```

void projet.lasalle84.darts.ActiviteAjouterJoueur.recupererWidget ( )

```

Définition à la ligne 49 du fichier [ActiviteAjouterJoueur.java](#).

Référencé par [projet.lasalle84.darts.ActiviteAjouterJoueur.onCreate\(\)](#).

```

00050     {
00051         Log.d(TAG, "recupererWidget()");
00052         boutonValider = (Button) findViewById(R.id.bouton);
00053         saisir = (EditText) findViewById(R.id.Nom);
00054     }

```

## 7.1.3 Documentation des données membres

## 7.1.3.1 boutonValider

```
Button projet.lasalle84.darts.ActiviteAjouterJoueur.boutonValider [private]
```

Bouton valider.

Définition à la ligne 26 du fichier [ActiviteAjouterJoueur.java](#).

## 7.1.3.2 saisir

```
EditText projet.lasalle84.darts.ActiviteAjouterJoueur.saisir [private]
```

Zone de saisir du nom du joueur.

Définition à la ligne 27 du fichier [ActiviteAjouterJoueur.java](#).

## 7.1.3.3 TAG

```
final String projet.lasalle84.darts.ActiviteAjouterJoueur.TAG = "IHMAjouterJoueur" [static], [private]
```

le TAG de la classe pour les logs

Définition à la ligne 25 du fichier [ActiviteAjouterJoueur.java](#).

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

— [ActiviteAjouterJoueur.java](#)

## 7.2 Référence de la classe projet.lasalle84.darts.ActiviteCreerPartie

Graphe de collaboration de projet.lasalle84.darts.ActiviteCreerPartie :

projet.lasalle84.darts.Activite CreerPartie
- Adaptateur - boutonAjouterJoueur - boutonLancerPartie - checkBoxAfficheRegle - listViewJoueur - modeDeJeu - TAG
+ LancerPartie() + onActivityResult() + onClick() # onCreate() - initialiserWidgets() - recupererWidgets()

### Fonctions membres publiques

- void [LancerPartie](#) ()  
*Méthode pour lancer la partie.*
- void [onActivityResult](#) (int requestCode, int resultCode, Intent data)  
*Méthode qui permet de récupérer le nom du joueur.*
- void [onClick](#) (View element)  
*Méthode appelée lors d'un clic sur un widget de l'activité*

### Fonctions membres protégées

- void [onCreate](#) (Bundle savedInstanceState)  
*Méthode appelée à la création de l'activité*

### Fonctions membres privées

- void [initialiserWidgets](#) ()  
*Méthode pour initialiser les Widget.*
- void [recupererWidgets](#) ()  
*Méthode pour récupérer les Widget.*

### Attributs privés

- ArrayAdapter< String > [Adaptateur](#)  
*Adapter pour afficher sur ListView.*
- Button [boutonAjouterJoueur](#)  
*Bouton Ajouter un joueur.*
- Button [boutonLancerPartie](#)  
*Bouton pour lancer une partie.*
- CheckBox [checkBoxAfficheRegle](#)
- ListView [listViewJoueur](#)  
*ListView avec les nom des joueurs.*
- Spinner [modeDeJeu](#)  
*Menu déroulant pour choisir le mode de jeu.*

### Attributs privés statiques

- static final String [TAG](#) = "IHMCreerPartie"  
*le TAG de la classe pour les logs*

#### 7.2.1 Description détaillée

Définition à la ligne [22](#) du fichier [ActiviteCreerPartie.java](#).

#### 7.2.2 Documentation des fonctions membres

### 7.2.2.1 initialiserWidgets()

```
projet.lasalle84.darts.ActiviteCreerPartie.initialiserWidgets ( ) [private]
```

Méthode pour initialiser les Widget.

Définition à la ligne 130 du fichier [ActiviteCreerPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteCreerPartie.onCreate\(\)](#).

```
00131     {
00132         Log.d(TAG, "initialiserWidgets()");
00133         boutonAjouterJoueur.setOnClickListener(this);
00134         boutonLancerPartie.setOnClickListener(this);
00135         Adaptateur = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
00136         listViewJoueur.setAdapter(Adaptateur);
00137     }
```

### 7.2.2.2 LancerPartie()

```
projet.lasalle84.darts.ActiviteCreerPartie.LancerPartie ( )
```

Méthode pour lancer la partie.

Définition à la ligne 79 du fichier [ActiviteCreerPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteCreerPartie.onClick\(\)](#).

```
00080     {
00081         Log.d(TAG, "LancerPartie()");
00082         Vector<Joueur>mesJoueurs = new Vector<>();
00083         if(Adaptateur.getCount() == 0)
00084         {
00085             for (int i = 0; i < 4; i++)
00086             {
00087                 mesJoueurs.add(new Joueur("Joueur" + i));
00088                 Log.d(TAG, "Joueur" + i + " = " + "Joueur" + i);
00089             }
00090         }
00091         else
00092         {
00093             for (int i = 0; i < Adaptateur.getCount(); i++)
00094             {
00095                 mesJoueurs.add(new Joueur(Adaptateur.getItem(i)));
00096                 Log.d(TAG, "Joueur" + i + " = " + Adaptateur.getItem(i));
00097             }
00098         }
00099
00100         Intent intent = new Intent(ActiviteCreerPartie.this, ActivitePartie.class);
00101         intent.putExtra("TypeMode", modeDeJeu.getSelectedItemPosition());
00102         Log.d(TAG, "TypeMode = " + modeDeJeu.getSelectedItemPosition());
00103         intent.putExtra("LesJoueurs", mesJoueurs);
00104         intent.putExtra("AfficheRegle", checkBoxAfficheRegle.isChecked());
00105         startActivity(intent);
00106         super.finish();
00107     }
```

### 7.2.2.3 onActivityResult()

```

projet.lasalle84.darts.ActiviteCreerPartie.onActivityResult (
    int requestCode,
    int resultCode,
    Intent data )

```

Méthode qui permet de récupérer le nom du joueur.

Définition à la ligne 144 du fichier [ActiviteCreerPartie.java](#).

```

00144                                     {
00145     super.onActivityResult(requestCode, resultCode, data);
00146     if (requestCode == 1)
00147     {
00148         if (resultCode == RESULT_OK) {
00149             String nomJoueur = data.getStringExtra("Joueur");
00150             Log.d(TAG,"nom du joueur: " + nomJoueur);
00151             Adaptateur.add(nomJoueur);
00152         }
00153     }
00154 }

```

### 7.2.2.4 onClick()

```

projet.lasalle84.darts.ActiviteCreerPartie.onClick (
    View element )

```

Méthode appelée lors d'un clic sur un widget de l'activité

#### Paramètres

<i>element</i>	le widget cliqué
----------------	------------------

Définition à la ligne 57 du fichier [ActiviteCreerPartie.java](#).

Références [projet.lasalle84.darts.ActiviteCreerPartie.LancerPartie\(\)](#).

```

00058     {
00059         Log.d(TAG, "onClick()");
00060
00061         if(element == boutonAjouterJoueur)
00062         {
00063             Log.d(TAG, "clic boutonAjouterJoueur");
00064             Intent intent = new Intent(ActiviteCreerPartie.this, ActiviteAjouterJoueur.class);
00065             startActivityForResult(intent,1);
00066         }
00067         else if (element == boutonLancerPartie)
00068         {
00069             Log.d(TAG,"clic boutonLancerPartie");
00070             LancerPartie();
00071         }
00072     }
00073 }

```

### 7.2.2.5 onCreate()

```

projet.lasalle84.darts.ActiviteCreerPartie.onCreate (
    Bundle savedInstanceState ) [protected]

```

Méthode appelée à la création de l'activité

## Paramètres

<code>savedInstanceState</code>	
---------------------------------	--

Définition à la ligne 40 du fichier [ActiviteCreerPartie.java](#).

Références [projet.lasalle84.darts.ActiviteCreerPartie.initialiserWidgets\(\)](#), et [projet.lasalle84.darts.ActiviteCreerPartie.recupererWidgets\(\)](#).

```
00040         {
00041             Log.d(TAG, "onCreate()");
00042             super.onCreate(savedInstanceState);
00043             setContentView(R.layout.activity_creer_partie);
00044             recupererWidgets();
00045             initialiserWidgets();
00046         }
00047     }
```

## 7.2.2.6 recupererWidgets()

```
projet.lasalle84.darts.ActiviteCreerPartie.recupererWidgets ( ) [private]
```

Méthode pour récupérer les Widget.

Définition à la ligne 113 du fichier [ActiviteCreerPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteCreerPartie.onCreate\(\)](#).

```
00114     {
00115         Log.d(TAG, "recupererWidgets()");
00116
00117         boutonAjouterJoueur = (Button) findViewById(R.id.AjouterJoueur);
00118         modeDeJeu = (Spinner) findViewById(R.id.modeJeu);
00119         listViewJoueur = (ListView) findViewById(R.id.ListeJoueur);
00120         boutonLancerPartie = (Button) findViewById(R.id.boutonLancerPartie);
00121         checkBoxAfficheRegle = (CheckBox) findViewById(R.id.afficheRegle);
00122
00123     }
00124 }
```

## 7.2.3 Documentation des données membres

## 7.2.3.1 Adaptateur

```
ArrayAdapter<String> projet.lasalle84.darts.ActiviteCreerPartie.Adaptateur [private]
```

Adapter pour afficher sur ListView.

Définition à la ligne 30 du fichier [ActiviteCreerPartie.java](#).

### 7.2.3.2 boutonAjouterJoueur

```
Button projet.lasalle84.darts.ActiviteCreerPartie.boutonAjouterJoueur [private]
```

Bouton Ajouter un joueur.

Définition à la ligne 27 du fichier [ActiviteCreerPartie.java](#).

### 7.2.3.3 boutonLancerPartie

```
Button projet.lasalle84.darts.ActiviteCreerPartie.boutonLancerPartie [private]
```

Bouton pour lancer une partie.

Définition à la ligne 31 du fichier [ActiviteCreerPartie.java](#).

### 7.2.3.4 checkBoxAfficheRegle

```
CheckBox projet.lasalle84.darts.ActiviteCreerPartie.checkBoxAfficheRegle [private]
```

Définition à la ligne 32 du fichier [ActiviteCreerPartie.java](#).

### 7.2.3.5 listViewJoueur

```
ListView projet.lasalle84.darts.ActiviteCreerPartie.listViewJoueur [private]
```

ListView avec les nom des joueurs.

Définition à la ligne 29 du fichier [ActiviteCreerPartie.java](#).

### 7.2.3.6 modeDeJeu

```
Spinner projet.lasalle84.darts.ActiviteCreerPartie.modeDeJeu [private]
```

Menu déroulant pour choisir le mode de jeu.

Définition à la ligne 28 du fichier [ActiviteCreerPartie.java](#).

### 7.2.3.7 TAG

```
final String projet.lasalle84.darts.ActiviteCreerPartie.TAG = "IHMCreerPartie" [static], [private]
```

le TAG de la classe pour les logs

Définition à la ligne 25 du fichier [ActiviteCreerPartie.java](#).

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

— [ActiviteCreerPartie.java](#)

## 7.3 Référence de la classe projet.lasalle84.darts.ActiviteFinPartie

Graphe de collaboration de projet.lasalle84.darts.ActiviteFinPartie :

projet.lasalle84.darts.Activite FinPartie
- adapteur - classement - lesJoueur - nomGagnant - partieFini - TAG - texteGagnant
+ afficherJoueurs() + initialiserWidgets() + onClick() + recupererInfomationPartie() + recupererWidgets() + trierLesJoueurs() # onCreate()

## Fonctions membres publiques

- void [afficherJoueurs](#) ()  
*Méthode appelée à la création de l'activité*
- void [initialiserWidgets](#) ()  
*Méthode permet initialiser les widgets.*
- void [onClick](#) (View element)  
*Méthode appelée à la création de l'activité*
- void [recupererInfomationPartie](#) ()  
*Méthode appelée à la création de l'activité*
- void [recupererWidgets](#) ()  
*Méthode permet récupérer les widgets du layout.*
- void [trierLesJoueurs](#) ()  
*Méthode permet de trier les joueur par rapport au score.*

## Fonctions membres protégées

- void [onCreate](#) (Bundle savedInstanceState)  
*Méthode appelée à la création de l'activité*

## Attributs privés

- ArrayAdapter< String > [adapteur](#) = null
- ListView [classement](#) = null
- ArrayList< Joueur > [lesJoueur](#) = null
- String [nomGagnant](#) = null
- Button [partieFini](#) = null
- final String [TAG](#) = "IHMFinPartie"
- TextView [texteGagnant](#) = null

### 7.3.1 Description détaillée

Définition à la ligne 22 du fichier [ActiviteFinPartie.java](#).

### 7.3.2 Documentation des fonctions membres

#### 7.3.2.1 afficherJoueurs()

`projet.lasalle84.darts.ActiviteFinPartie.afficherJoueurs ( )`

Méthode appelée à la création de l'activité

Définition à la ligne 127 du fichier [ActiviteFinPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteFinPartie.onCreate\(\)](#).

```
00128     {
00129         Log.d(TAG, "afficherJoueurs()");
00130         adapteur = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
00131         for (int i = 0; i < lesJoueur.size(); i++)
00132         {
00133             adapteur.add(lesJoueur.get(i).getNom() + " : " +
lesJoueur.get(i).getScore());
00134         }
00135         classement.setAdapter(adapteur);
00136
00137         texteGagnant.setText(nomGagnant + " a gagné la partie !");
00138     }
```

#### 7.3.2.2 initialiserWidgets()

`projet.lasalle84.darts.ActiviteFinPartie.initialiserWidgets ( )`

Méthode permet initialiser les widgets.

Définition à la ligne 91 du fichier [ActiviteFinPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteFinPartie.onCreate\(\)](#).

```
00092     {
00093         Log.d(TAG, "initialiserWidgets()");
00094         partieFini.setOnClickListener(this);
00095     }
```

#### 7.3.2.3 onClick()

`projet.lasalle84.darts.ActiviteFinPartie.onClick (`  
`View element )`

Méthode appelée à la création de l'activité

Définition à la ligne 62 du fichier [ActiviteFinPartie.java](#).

```
00063     {
00064         Log.d(TAG, "onClick() " + element.toString());
00065
00066         if(element == partieFini)
00067         {
00068             super.finish();
00069         }
00070     }
```

#### 7.3.2.4 onCreate()

`projet.lasalle84.darts.ActiviteFinPartie.onCreate (`  
`Bundle savedInstanceState ) [protected]`

Méthode appelée à la création de l'activité

## Paramètres

<code>savedInstanceState</code>	
---------------------------------	--

Définition à la ligne 45 du fichier [ActiviteFinPartie.java](#).

Références [projet.lasalle84.darts.ActiviteFinPartie.afficherJoueurs\(\)](#), [projet.lasalle84.darts.ActiviteFinPartie.initialiserWidgets\(\)](#), [projet.lasalle84.darts.ActiviteFinPartie.recupererInfomationPartie\(\)](#), [projet.lasalle84.darts.ActiviteFinPartie.recupererWidgets\(\)](#), et [projet.lasalle84.darts.ActiviteFinPartie.trierLesJoueurs\(\)](#).

```

00045         {
00046             super.onCreate(savedInstanceState);
00047             Log.d(TAG, "onCreate()");
00048             setContentView(R.layout.activite_fin_partie);
00049             recupererWidgets();
00050             initialiserWidgets();
00051             recupererInfomationPartie();
00052             trierLesJoueurs();
00053             afficherJoueurs();
00054         }

```

## 7.3.2.5 recupererInfomationPartie()

```
projet.lasalle84.darts.ActiviteFinPartie.recupererInfomationPartie ( )
```

Méthode appelée à la création de l'activité

Définition à la ligne 102 du fichier [ActiviteFinPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteFinPartie.onCreate\(\)](#).

```

00103     {
00104         Log.d(TAG, "recupererInfomationPartie()");
00105         lesJoueur = (ArrayList<Joueur>) getIntent().getSerializableExtra("joueurs");
00106         nomGagnant = getIntent().getStringExtra("gagnant");
00107     }
00108 }

```

## 7.3.2.6 recupererWidgets()

```
projet.lasalle84.darts.ActiviteFinPartie.recupererWidgets ( )
```

Méthode permet récupérer les widgets du layout.

Définition à la ligne 77 du fichier [ActiviteFinPartie.java](#).

Référencé par [projet.lasalle84.darts.ActiviteFinPartie.onCreate\(\)](#).

```

00078     {
00079         Log.d(TAG, "recupererWidgets()");
00080         texteGagnant = (TextView) findViewById(R.id.texteGagnant);
00081         classement = (ListView) findViewById(R.id.ResultaPartie);
00082         partieFini = (Button) findViewById(R.id.boutonFin);
00083     }
00084 }

```

### 7.3.2.7 trierLesJoueurs()

```
projet.lasalle84.darts.ActiviteFinPartie.trierLesJoueurs ( )
```

Méthode permet de trier les joueur par rapport au score.

Définition à la ligne 115 du fichier [ActiviteFinPartie.java](#).

Références [projet.lasalle84.darts.Joueur.getScore\(\)](#).

Référencé par [projet.lasalle84.darts.ActiviteFinPartie.onCreate\(\)](#).

```
00116     {
00117         Log.d(TAG, "trierLesJoueurs ( )");
00118         lesJoueur.sort(Comparator.comparingInt(Joueur::getScore));
00119     }
00120 }
```

## 7.3.3 Documentation des données membres

### 7.3.3.1 adaptateur

```
ArrayAdapter<String> projet.lasalle84.darts.ActiviteFinPartie.adaptateur = null [private]
```

Définition à la ligne 36 du fichier [ActiviteFinPartie.java](#).

### 7.3.3.2 classement

```
ListView projet.lasalle84.darts.ActiviteFinPartie.classement = null [private]
```

Définition à la ligne 32 du fichier [ActiviteFinPartie.java](#).

### 7.3.3.3 lesJoueur

```
ArrayList<Joueur> projet.lasalle84.darts.ActiviteFinPartie.lesJoueur = null [private]
```

Définition à la ligne 34 du fichier [ActiviteFinPartie.java](#).

### 7.3.3.4 nomGagnant

```
String projet.lasalle84.darts.ActiviteFinPartie.nomGagnant = null [private]
```

Définition à la ligne 35 du fichier [ActiviteFinPartie.java](#).

### 7.3.3.5 partieFini

```
Button projet.lasalle84.darts.ActiviteFinPartie.partieFini = null [private]
```

Définition à la ligne 33 du fichier [ActiviteFinPartie.java](#).

### 7.3.3.6 TAG

```
final String projet.lasalle84.darts.ActiviteFinPartie.TAG = "IHMFinPartie" [private]
```

Constantes

Définition à la ligne 27 du fichier [ActiviteFinPartie.java](#).

### 7.3.3.7 texteGagnant

```
TextView projet.lasalle84.darts.ActiviteFinPartie.texteGagnant = null [private]
```

Attributs

Définition à la ligne 31 du fichier [ActiviteFinPartie.java](#).

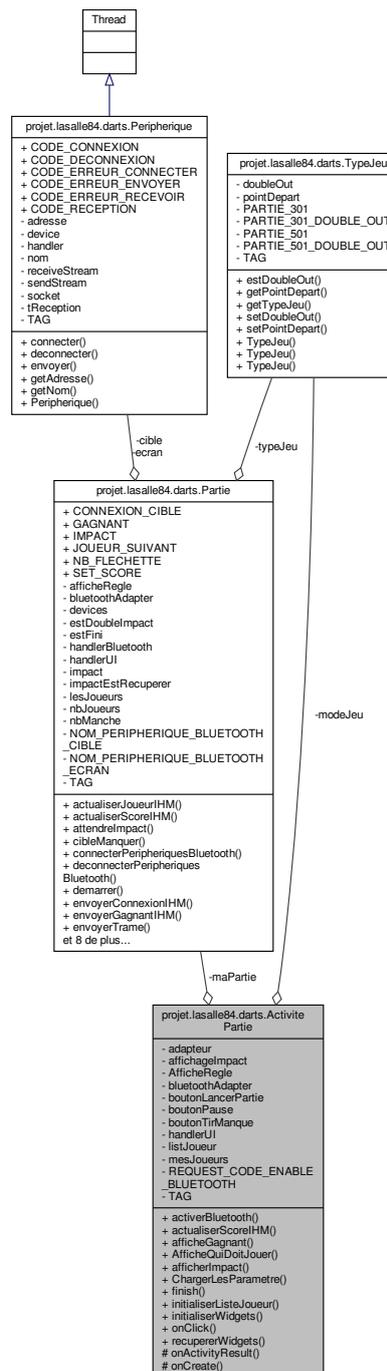
La documentation de cette classe a été générée à partir du fichier suivant :

— [ActiviteFinPartie.java](#)

## 7.4 Référence de la classe projet.lasalle84.darts.ActivitePartie

Déclaration de la classe [ActivitePartie](#).

Graphe de collaboration de projet.lasalle84.darts.ActivitePartie :



## Fonctions membres publiques

- void [activerBluetooth](#) ()
- void [actualiserScoreIHM](#) (String joueur, int score)  
*Méthode qui actualise le score du joueur sur IHM.*
- void [afficheGagnant](#) (String gagnant)
- void [AfficheQuiDoitJouer](#) (String joueur)  
*Méthode qui affiche qui doit jouer.*
- void [afficherImpact](#) (int typelImpact, int numeroImpact)  
*Méthode qui affiche l'impact de la volley actuelle.*
- void [ChargerLesParametre](#) ()

- Méthode qui permet charger des joueurs et les paramètre de la partie.*
- void [finish](#) ()
- void [initialiserListeJoueur](#) ()
- void [initialiserWidgets](#) ()
  - Récupère les widgets du layout de l'activité*
- void [onClick](#) (View element)
  - Méthode appelée lors d'un clic sur un widget de l'activité*
- void [recupererWidgets](#) ()
  - Récupère les widgets du layout de l'activité*

#### Fonctions membres protégées

- void [onActivityResult](#) (int requestCode, int resultCode, Intent data)
  - Méthode qui permet d'avoir le retour d'une activité*
- void [onCreate](#) (Bundle savedInstanceState)
  - Méthode appelée à la création de l'activité*

#### Attributs privés

- ArrayAdapter< String > [adapteur](#) = null
  - Adapteur qui affiche quelle joueur doit jouer et score.*
- TextView [affichageImpact](#) = null
  - Texte qui affiche la volley actuelle.*
- boolean [AfficheRegle](#)
- BluetoothAdapter [bluetoothAdapter](#) = null
  - Bluetooth Adapteur.*
- Button [boutonLancerPartie](#) = null
  - Bouton pour lancer la partie.*
- Button [boutonPause](#) = null
  - Bouton Pause.*
- Button [boutonTirManque](#) = null
  - Le bouton si la cible a était manquée.*
- final Handler [handlerUI](#)
- ListView [listJoueur](#) = null
  - Listview.*
- [Partie maPartie](#) = null
  - La partie.*
- ArrayList< [Joueur](#) > [mesJoueurs](#) = null
  - liste de [Joueur](#)*
- [TypeJeu modeJeu](#) = null
  - Object Type de Jeu.*

#### Attributs privés statiques

- static final int [REQUEST\\_CODE\\_ENABLE\\_BLUETOOTH](#) = 0
  - Code que le bluetooth est activé*
- static final String [TAG](#) = "IHMPartie"
  - Tag pour Log.*

#### 7.4.1 Description détaillée

Déclaration de la classe [ActivitePartie](#).

Définition à la ligne 32 du fichier [ActivitePartie.java](#).

#### 7.4.2 Documentation des fonctions membres

### 7.4.2.1 activerBluetooth()

```
void projet.lasalle84.darts.ActivitePartie.activerBluetooth ( )
```

Définition à la ligne 135 du fichier [ActivitePartie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onCreate\(\)](#).

```
00136     {
00137         bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
00138         if (bluetoothAdapter == null)
00139         {
00140             Toast.makeText(getApplicationContext(), "Bluetooth non activé !", Toast.LENGTH_SHORT).show();
00141         }
00142         else
00143         {
00144             if (!bluetoothAdapter.isEnabled())
00145             {
00146                 Toast.makeText(getApplicationContext(), "Bluetooth non activé !", Toast.LENGTH_SHORT).show(
00147             );
00148                 Intent activeBlueTooth = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
00149                 startActivityForResult (activeBlueTooth,
00150                 REQUEST_CODE_ENABLE_BLUETOOTH);
00151             }
00152             else
00153             {
00154                 Toast.makeText(getApplicationContext(), "Bluetooth activé", Toast.LENGTH_SHORT).show();
00155             }
00156         }
00157     }
```

### 7.4.2.2 actualiserScoreIHM()

```
projet.lasalle84.darts.ActivitePartie.actualiserScoreIHM (
    String joueur,
    int score )
```

Méthode qui actualise le score du joueur sur IHM.

Définition à la ligne 273 du fichier [ActivitePartie.java](#).

```
00274     {
00275         Log.d(TAG, "actualiserScoreIHM()");
00276         int idJoueur = 0;
00277
00278         for (int i = 0; i < mesJoueurs.size(); i++)
00279         {
00280             if (mesJoueurs.get(i).getNom().equals(joueur))
00281             {
00282                 idJoueur = i;
00283             }
00284         }
00285         mesJoueurs.get(idJoueur).setScore(score);
00286         String chaineCaractere = mesJoueurs.get(idJoueur).getNom() + ": " +
00287         mesJoueurs.get(idJoueur).getScore();
00288         adapteur.remove(adapteur.getItem(idJoueur));
00289         adapteur.insert(chaineCaractere, idJoueur);
00290     }
```

## 7.4.2.3 afficheGagnant()

```
void projet.lasalle84.darts.ActivitePartie.afficheGagnant (
    String gagnant )
```

Définition à la ligne 358 du fichier [ActivitePartie.java](#).

Références [projet.lasalle84.darts.ActivitePartie.finish\(\)](#).

```
00359     {
00360         Log.d(TAG, "afficheGagnant()");
00361         Intent IHMGagnant = new Intent(ActivitePartie.this, ActiviteFinPartie.class);
00362         List<Joueur>lesJoueur = new ArrayList<Joueur>();
00363         for (int i = 0; i < mesJoueurs.size(); i++)
00364         {
00365             if (mesJoueurs.get(i).getNom().equals(gagnant))
00366             {
00367                 IHMGagnant.putExtra("gagnant", mesJoueurs.get(i).getNom());
00368             }
00369             else
00370             {
00371                 lesJoueur.add(mesJoueurs.get(i));
00372             }
00373
00374             IHMGagnant.putExtra("joueurs", (Serializable) lesJoueur);
00375             startActivity(IHMGagnant);
00376             finish();
00377         }
00378     }
```

## 7.4.2.4 AfficheQuiDoitJouer()

```
projet.lasalle84.darts.ActivitePartie.AfficheQuiDoitJouer (
    String joueur )
```

Méthode qui affiche qui doit jouer.

Définition à la ligne 297 du fichier [ActivitePartie.java](#).

```
00298     {
00299         Log.d(TAG, "AfficheQuiDoitJouer()");
00300         String chaineCaractere;
00301         for (int i = 0; i < mesJoueurs.size(); i++)
00302         {
00303             if (mesJoueurs.get(i).getNom().equals(joueur))
00304             {
00305                 chaineCaractere = "-> " + mesJoueurs.get(i).getNom() + ": " +
mesJoueurs.get(i).getScore();
00306             }
00307             else {
00308                 chaineCaractere = mesJoueurs.get(i).getNom() + ": " +
mesJoueurs.get(i).getScore();
00309             }
00310             adapteur.remove(adapteur.getItem(i));
00311             adapteur.insert(chaineCaractere, i);
00312         }
00313     }
00314     affichageImpact.setText("");
00315 }
```

### 7.4.2.5 afficherImpact()

```
projet.lasalle84.darts.ActivitePartie.afficherImpact (
    int typeImpact,
    int numeroImpact )
```

Méthode qui affiche l'impact de la volley actuelle.

Définition à la ligne 323 du fichier [ActivitePartie.java](#).

```
00324     {
00325         Log.d(TAG, "afficherImpact()");
00326         String lettreTypeImpact = new String();
00327         switch (typeImpact)
00328         {
00329             case 0:
00330                 lettreTypeImpact = "MISS";
00331                 break;
00332             case 1:
00333                 lettreTypeImpact = "S";
00334                 break;
00335             case 2:
00336                 lettreTypeImpact = "D";
00337                 break;
00338             case 3:
00339                 lettreTypeImpact = "T";
00340                 break;
00341         }
00342         if (lettreTypeImpact == "MISS")
00343         {
00344             affichageImpact.setText(affichageImpact.getText().toString().
concat("MISS "));
00345         }
00346         else
00347         {
00348             affichageImpact.setText(affichageImpact.getText().toString().
concat(lettreTypeImpact + Integer.toString(numeroImpact) + " "));
00349         }
00350     }
```

### 7.4.2.6 ChargerLesParametre()

```
projet.lasalle84.darts.ActivitePartie.ChargerLesParametre ( )
```

Méthode qui permet charger des joueurs et les paramètre de la partie.

Définition à la ligne 79 du fichier [ActivitePartie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onCreate\(\)](#).

```
00080     {
00081         mesJoueurs = new ArrayList<Joueur>();
00082         mesJoueurs = (ArrayList<Joueur>) getIntent().getSerializableExtra("LesJoueurs");
00083         Iterator<Joueur> it = mesJoueurs.iterator();
00084
00085         while(it.hasNext())
00086         {
00087             Log.d(TAG, "le joueur "+ it.next().getNom() + "est chargé");
00088         }
00089         modeJeu = new TypeJeu(getIntent().getIntExtra("TypeMode", 0));
00090         AfficheRegle = getIntent().getBooleanExtra("AfficheRegle", true);
00091     }
```

#### 7.4.2.7 finish()

```
void projet.lasalle84.darts.ActivitePartie.finish ( )
```

Définition à la ligne 381 du fichier [ActivitePartie.java](#).

Références [projet.lasalle84.darts.Partie.deconnecterPeripheriquesBluetooth\(\)](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.afficheGagnant\(\)](#).

```
00382     {
00383         super.finish();
00384         maPartie.deconnecterPeripheriquesBluetooth();
00385     }
```

#### 7.4.2.8 initialiserListeJoueur()

```
void projet.lasalle84.darts.ActivitePartie.initialiserListeJoueur ( )
```

Définition à la ligne 256 du fichier [ActivitePartie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onCreate\(\)](#).

```
00257     {
00258         Log.d(TAG, "initialiserListeJoueur()");
00259         adapteur = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
00260         for (int i = 0; i < mesJoueurs.size(); i++)
00261         {
00262             adapteur.add(mesJoueurs.get(i).getNom());
00263         }
00264         listJoueur.setAdapter(adapteur);
00265     }
```

#### 7.4.2.9 initialiserWidgets()

```
projet.lasalle84.darts.ActivitePartie.initialiserWidgets ( )
```

Récupère les widgets du layout de l'activité

Définition à la ligne 180 du fichier [ActivitePartie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onCreate\(\)](#).

```
00181     {
00182         Log.d(TAG, "initialiserWidgets()");
00183         boutonTirManque.setOnClickListener(this);
00184         boutonTirManque.setVisibility(View.INVISIBLE);
00185         boutonLancerPartie.setVisibility(View.INVISIBLE);
00186         boutonLancerPartie.setOnClickListener(this);
00187         boutonPause.setVisibility(View.INVISIBLE); // Fonction Pause non-fonctionnel
00188         boutonPause.setOnClickListener(this);
00189     }
```

### 7.4.2.10 onActivityResult()

```
projet.lasalle84.darts.ActivitePartie.onActivityResult (
    int requestCode,
    int resultCode,
    Intent data ) [protected]
```

Méthode qui permet d'avoir le retour d'une activité

Définition à la ligne 198 du fichier [ActivitePartie.java](#).

```
00199     {
00200         Log.d(TAG, "onActivityResult ()");
00201         super.onActivityResult(requestCode, resultCode, data);
00202         if (requestCode != REQUEST_CODE_ENABLE_BLUETOOTH)
00203             return;
00204         if (resultCode == RESULT_OK)
00205         {
00206             Toast.makeText(getApplicationContext(), "Bluetooth activé", Toast.LENGTH_SHORT).show();
00207         }
00208         else
00209         {
00210             Toast.makeText(getApplicationContext(), "Bluetooth non activé !", Toast.LENGTH_SHORT).show();
00211         }
00212     }
```

### 7.4.2.11 onClick()

```
projet.lasalle84.darts.ActivitePartie.onClick (
    View element )
```

Méthode appelée lors d'un clic sur un widget de l'activité

#### Paramètres

<i>element</i>	le widget cliqué
----------------	------------------

Définition à la ligne 100 du fichier [ActivitePartie.java](#).

Références [projet.lasalle84.darts.Partie.cibleManquer\(\)](#), et [projet.lasalle84.darts.Partie.demarrer\(\)](#).

```
00101     {
00102         Log.d(TAG, "onClick() " + element.toString());
00103
00104         if (element == boutonLancerPartie)
00105         {
00106             new Thread()
00107             {
00108                 @Override
00109                 public void run() {
00110                     super.run();
00111                     maPartie.demarrer();
00112                 }
00113             }.start();
00114             boutonLancerPartie.setVisibility(View.INVISIBLE);
00115             boutonTirManque.setVisibility(View.VISIBLE);
00116         }
00117         else if (element == boutonTirManque)
00118         {
00119             maPartie.cibleManquer();
00120         }
00121         else if (element == boutonPause)
00122         {
00123             if (boutonPause.getText().equals("Pause"))
00124             {
00125                 boutonPause.setText("Reprendre");
00126             }
00127         }
00128     }
```

```

00126         }
00127         else
00128         {
00129             boutonPause.setText("Pause");
00130         }
00131     }
00132
00133
00134 }

```

#### 7.4.2.12 onCreate()

```

projet.lasalle84.darts.ActivitePartie.onCreate (
    Bundle savedInstanceState ) [protected]

```

Méthode appelée à la création de l'activité

##### Paramètres

<i>savedInstanceState</i>
---------------------------

Définition à la ligne 61 du fichier [ActivitePartie.java](#).

Références [projet.lasalle84.darts.ActivitePartie.activerBluetooth\(\)](#), [projet.lasalle84.darts.ActivitePartie.ChargerLesParametre\(\)](#), [projet.lasalle84.darts.ActivitePartie.handlerUI](#), [projet.lasalle84.darts.ActivitePartie.initialiserListeJoueur\(\)](#), [projet.lasalle84.darts.ActivitePartie.initialiserWidgets\(\)](#), et [projet.lasalle84.darts.ActivitePartie.recupererWidgets\(\)](#).

```

00062     {
00063         Log.d(TAG, "onCreate()");
00064         super.onCreate(savedInstanceState);
00065         setContentView(R.layout.activity_partie);
00066         recupererWidgets();
00067         initialiserWidgets();
00068         activerBluetooth();
00069         ChargerLesParametre();
00070         initialiserListeJoueur();
00071         maPartie = new Partie(mesJoueurs, modeJeu,
AfficheRegle, bluetoothAdapter, handlerUI);
00072     }

```

#### 7.4.2.13 recupererWidgets()

```

projet.lasalle84.darts.ActivitePartie.recupererWidgets ( )

```

Récupère les widgets du layout de l'activité

Définition à la ligne 164 du fichier [ActivitePartie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onCreate\(\)](#).

```

00165     {
00166         Log.d(TAG, "recupererWidgets()");
00167         boutonTirManque = (Button) findViewById(R.id.TirManque);
00168         boutonLancerPartie = (Button) findViewById(R.id.LancerPartie);
00169         listJoueur = (ListView) findViewById(R.id.MesJoueur);
00170         affichageImpact = (TextView) findViewById(R.id.Impact);
00171         boutonPause = (Button) findViewById(R.id.Pause);
00172     }

```

### 7.4.3 Documentation des données membres

#### 7.4.3.1 adaptateur

```
ArrayAdapter<String> projet.lasalle84.darts.ActivitePartie.adaptateur = null [private]
```

Adaptateur qui affiche quelle joueur doit jouer et score.

Définition à la ligne 48 du fichier [ActivitePartie.java](#).

#### 7.4.3.2 affichagelmpact

```
TextView projet.lasalle84.darts.ActivitePartie.affichageImpact = null [private]
```

Texte qui affiche la volley actuelle.

Définition à la ligne 50 du fichier [ActivitePartie.java](#).

#### 7.4.3.3 AfficheRegle

```
boolean projet.lasalle84.darts.ActivitePartie.AfficheRegle [private]
```

Définition à la ligne 52 du fichier [ActivitePartie.java](#).

#### 7.4.3.4 bluetoothAdapter

```
BluetoothAdapter projet.lasalle84.darts.ActivitePartie.bluetoothAdapter = null [private]
```

Bluetooth Adaptateur.

Définition à la ligne 45 du fichier [ActivitePartie.java](#).

#### 7.4.3.5 boutonLancerPartie

```
Button projet.lasalle84.darts.ActivitePartie.boutonLancerPartie = null [private]
```

Bouton pour lancer la partie.

Définition à la ligne 43 du fichier [ActivitePartie.java](#).

#### 7.4.3.6 boutonPause

```
Button projet.lasalle84.darts.ActivitePartie.boutonPause = null [private]
```

Bouton Pause.

Définition à la ligne 51 du fichier [ActivitePartie.java](#).

#### 7.4.3.7 boutonTirManque

```
Button projet.lasalle84.darts.ActivitePartie.boutonTirManque = null [private]
```

Le bouton si la cible a été manquée.

Attributs

Définition à la ligne 42 du fichier [ActivitePartie.java](#).

#### 7.4.3.8 handlerUI

```
final Handler projet.lasalle84.darts.ActivitePartie.handlerUI [private]
```

Définition à la ligne 214 du fichier [ActivitePartie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onCreate\(\)](#).

#### 7.4.3.9 listJoueur

```
ListView projet.lasalle84.darts.ActivitePartie.listJoueur = null [private]
```

Listview.

Définition à la ligne 49 du fichier [ActivitePartie.java](#).

#### 7.4.3.10 maPartie

```
Partie projet.lasalle84.darts.ActivitePartie.maPartie = null [private]
```

La partie.

Définition à la ligne 46 du fichier [ActivitePartie.java](#).

#### 7.4.3.11 mesJoueurs

```
ArrayList<Joueur> projet.lasalle84.darts.ActivitePartie.mesJoueurs = null [private]
```

liste de [Joueur](#)

Définition à la ligne [44](#) du fichier [ActivitePartie.java](#).

#### 7.4.3.12 modeJeu

```
TypeJeu projet.lasalle84.darts.ActivitePartie.modeJeu = null [private]
```

Object Type de Jeu.

Définition à la ligne [47](#) du fichier [ActivitePartie.java](#).

#### 7.4.3.13 REQUEST\_CODE\_ENABLE\_BLUETOOTH

```
final int projet.lasalle84.darts.ActivitePartie.REQUEST_CODE_ENABLE_BLUETOOTH = 0 [static], [private]
```

Code que le bluetooth est activé

Définition à la ligne [38](#) du fichier [ActivitePartie.java](#).

#### 7.4.3.14 TAG

```
final String projet.lasalle84.darts.ActivitePartie.TAG = "IHMPartie" [static], [private]
```

Tag pour Log.

Constantes

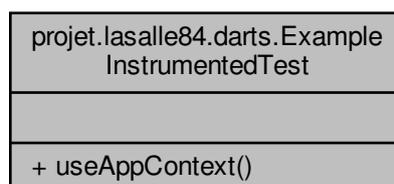
Définition à la ligne [37](#) du fichier [ActivitePartie.java](#).

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

— [ActivitePartie.java](#)

## 7.5 Référence de la classe projet.lasalle84.darts.ExampleInstrumentedTest

Graphe de collaboration de projet.lasalle84.darts.ExampleInstrumentedTest :



## Fonctions membres publiques

— void [useAppContext](#) ()

## 7.5.1 Description détaillée

Instrumented test, which will execute on an Android device.

## Voir également

[Testing documentation](#)

Définition à la ligne 19 du fichier [ExampleInstrumentedTest.java](#).

## 7.5.2 Documentation des fonctions membres

## 7.5.2.1 useAppContext()

```
void projet.lasalle84.darts.ExampleInstrumentedTest.useAppContext ( )
```

Définition à la ligne 21 du fichier [ExampleInstrumentedTest.java](#).

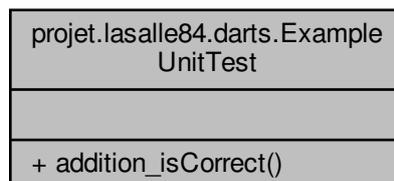
```
00021         {
00022         // Context of the app under test.
00023         Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
00024
00025         assertEquals("projet.lasalle84.darts", appContext.getPackageName());
00026     }
```

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

— [ExampleInstrumentedTest.java](#)

## 7.6 Référence de la classe projet.lasalle84.darts.ExampleUnitTest

Graphe de collaboration de projet.lasalle84.darts.ExampleUnitTest :



## Fonctions membres publiques

— void [addition\\_isCorrect](#) ()

### 7.6.1 Description détaillée

Exemple local unit test, which will execute on the development machine (host).

#### Voir également

[Testing documentation](#)

Définition à la ligne [12](#) du fichier [ExampleUnitTest.java](#).

### 7.6.2 Documentation des fonctions membres

#### 7.6.2.1 addition\_isCorrect()

```
void projet.lasalle84.darts.ExampleUnitTest.addition_isCorrect ( )
```

Définition à la ligne [14](#) du fichier [ExampleUnitTest.java](#).

```
00014         assertEquals(4, 2 + 2);           {
00015     }
00016 }
```

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

— [ExampleUnitTest.java](#)

## 7.7 Référence de la classe projet.lasalle84.darts.Joueur

Déclaration de la classe [Joueur](#).

Graphe de collaboration de projet.lasalle84.darts.Joueur :

projet.lasalle84.darts.Joueur
- estEliminer - nom - score - TAG
+ estEliminer() + getNom() + getScore() + Joueur() + retirerPoint() + setEstEliminer() + setNom() + setScore()

## Fonctions membres publiques

- boolean `estEliminer` ()
- String `getNom` ()  
*Accesseur get du nom du joueur.*
- int `getScore` ()  
*Accesseur get du score du joueur.*
- `Joueur` (String `nom`)  
*Constructeur de la classe `Joueur`.*
- boolean `retirerPoint` (int `scoreLancer`, `Partie` `maPartie`)  
*fonction qui permet de retirer le score et retourner si le score a été retiré*
- void `setEstEliminer` (boolean `estEliminer`)  
*Accesseur set si le joueur est eliminer.*
- void `setNom` (String `nom`)  
*Accesseur set du nom du joueur.*
- void `setScore` (int `score`)  
*Accesseur set du score du joueur.*

## Attributs privés

- boolean `estEliminer`
- String `nom`  
*Le nom du joueur.*
- int `score`  
*Le score du joueur.*

## Attributs privés statiques

- static final String `TAG` = "Joueur"

## 7.7.1 Description détaillée

Déclaration de la classe `Joueur`.

Définition à la ligne 18 du fichier `Joueur.java`.

## 7.7.2 Documentation des constructeurs et destructeur

7.7.2.1 `Joueur()`

```
projet.lasalle84.darts.Joueur.Joueur (
    String nom )
```

Constructeur de la classe `Joueur`.

## Paramètres

<i>nom</i>	le nom du joueur
------------	------------------

Définition à la ligne 31 du fichier `Joueur.java`.

Références `projet.lasalle84.darts.Joueur.nom`.

```
00032     {
00033         this.nom = nom;
00034         this.score = 0;
00035         estEliminer = false;
00036     }
```

### 7.7.3 Documentation des fonctions membres

#### 7.7.3.1 estEliminer()

`boolean projet.lasalle84.darts.Joueur.estEliminer ( )`

Définition à la ligne 119 du fichier `Joueur.java`.

Référencé par `projet.lasalle84.darts.Joueur.setEstEliminer()`.

```
00120     {
00121         return estEliminer;
00122     }
```

#### 7.7.3.2 getNom()

`projet.lasalle84.darts.Joueur.getNom ( )`

Accesseur get du nom du joueur.

**Renvoie**

String le nom du joueur

Définition à la ligne 44 du fichier `Joueur.java`.

Références `projet.lasalle84.darts.Joueur.nom`.

Référencé par `projet.lasalle84.darts.Partie.actualiserScoreIHM()`, `projet.lasalle84.darts.Partie.demarrer()`, `projet.lasalle84.darts.Partie.envoyerGagnantIHM()`, `projet.lasalle84.darts.Partie.initialiserPartie()`, et `projet.lasalle84.darts.Joueur.retirerPoint()`.

```
00045     {
00046         return nom;
00047     }
```

#### 7.7.3.3 getScore()

`projet.lasalle84.darts.Joueur.getScore ( )`

Accesseur get du score du joueur.

**Renvoie**

int le score du joueur

Définition à la ligne 66 du fichier `Joueur.java`.

Références `projet.lasalle84.darts.Joueur.score`.

Référencé par `projet.lasalle84.darts.Partie.demarrer()`, et `projet.lasalle84.darts.ActiviteFinPartie.trierLesJoueurs()`.

```
00067     {
00068         return score;
00069     }
```

#### 7.7.3.4 retirerPoint()

```
projet.lasalle84.darts.Joueur.retirerPoint (
    int scoreLancer,
    Partie maPartie )
```

fonction qui permet de retirer le score et retourner si le score a été retiré

## Paramètres

<code>scoreLancer</code>	à retirer
--------------------------	-----------

Définition à la ligne 88 du fichier `Joueur.java`.

Références [projet.lasalle84.darts.TypeJeu.estDoubleOut\(\)](#), [projet.lasalle84.darts.Joueur.getNom\(\)](#), et [projet.lasalle84.darts.Partie.<->getTypeJeu\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

```
00089     {
00090         if (this.score - scoreLancer >= 0)
00091         {
00092             this.score = this.score - scoreLancer;
00093             if (this.score == 1 && maPartie.getTypeJeu().estDoubleOut())
00094             {
00095                 Log.d(TAG, this.getNom() + " est éliminé");
00096                 estEliminer = true;
00097             }
00098             return true;
00099         }
00100         return false;
00101     }
```

## 7.7.3.5 setEstEliminer()

```
projet.lasalle84.darts.Joueur.setEstEliminer (
    boolean estEliminer )
```

Accesseur set si le joueur est eliminer.

Accesseur get si le joueur est eliminer.

## Paramètres

<code>estEliminer</code>	si le joueur est eliminer
--------------------------	---------------------------

Définition à la ligne 109 du fichier `Joueur.java`.

Références [projet.lasalle84.darts.Joueur.estEliminer\(\)](#).

```
00110     {
00111         this.estEliminer = estEliminer;
00112     }
```

## 7.7.3.6 setNom()

```
projet.lasalle84.darts.Joueur.setNom (
    String nom )
```

Accesseur set du nom du joueur.

### Paramètres

<i>nom</i>	le nom du joueur
------------	------------------

Définition à la ligne 55 du fichier [Joueur.java](#).

Références [projet.lasalle84.darts.Joueur.nom](#).

```
00056     {
00057         this.nom = nom;
00058     }
```

#### 7.7.3.7 setScore()

```
projet.lasalle84.darts.Joueur.setScore (
    int score )
```

Accesseur set du score du joueur.

### Paramètres

<i>score</i>	le score du joueur
--------------	--------------------

Définition à la ligne 77 du fichier [Joueur.java](#).

Références [projet.lasalle84.darts.Joueur.score](#).

Référencé par [projet.lasalle84.darts.Partie.initialiserPartie\(\)](#).

```
00077         {
00078             Log.d(TAG, this.nom + " setScore " + score);
00079             this.score = score;
00080         }
```

#### 7.7.4 Documentation des données membres

##### 7.7.4.1 estEliminer

```
boolean projet.lasalle84.darts.Joueur.estEliminer [private]
```

Définition à la ligne 23 du fichier [Joueur.java](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

## 7.7.4.2 nom

```
String projet.lasalle84.darts.Joueur.nom [private]
```

Le nom du joueur.

Définition à la ligne 21 du fichier [Joueur.java](#).

Référencé par [projet.lasalle84.darts.Joueur.getNom\(\)](#), [projet.lasalle84.darts.Joueur.Joueur\(\)](#), et [projet.lasalle84.darts.Joueur.set←Nom\(\)](#).

## 7.7.4.3 score

```
int projet.lasalle84.darts.Joueur.score [private]
```

Le score du joueur.

Définition à la ligne 22 du fichier [Joueur.java](#).

Référencé par [projet.lasalle84.darts.Joueur.getScore\(\)](#), et [projet.lasalle84.darts.Joueur.setScore\(\)](#).

## 7.7.4.4 TAG

```
final String projet.lasalle84.darts.Joueur.TAG = "Joueur" [static], [private]
```

Définition à la ligne 20 du fichier [Joueur.java](#).

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

## 7.8 Référence de la classe projet.lasalle84.darts.MainActivity

Déclaration de la classe [MainActivity](#).

Grphe de collaboration de projet.lasalle84.darts.MainActivity :

projet.lasalle84.darts.MainActivity
- boutonDemarrerPartie - boutonGererJoueurs - boutonVoirHistorique - TAG
+ onClick() # onCreate() - initialiserWidgets() - recupererWidgets()

### Fonctions membres publiques

- void [onClick](#) (View element)  
*Méthode appelée lors d'un clic sur un widget de l'activité*

### Fonctions membres protégées

- void [onCreate](#) (Bundle savedInstanceState)  
*Méthode appelée à la création de l'activité*

### Fonctions membres privées

- void [initialiserWidgets](#) ()  
*Récupère les widgets du layout de l'activité*
- void [recupererWidgets](#) ()  
*Récupère les widgets du layout de l'activité*

### Attributs privés

- Button [boutonDemarrerPartie](#)
- Button [boutonGererJoueurs](#)
- Button [boutonVoirHistorique](#)

### Attributs privés statiques

- static final String [TAG](#) = "IHMEcranPrincipal"  
*le TAG de la classe pour les logs*

## 7.8.1 Description détaillée

Déclaration de la classe [MainActivity](#).

Définition à la ligne 21 du fichier [MainActivity.java](#).

## 7.8.2 Documentation des fonctions membres

### 7.8.2.1 [initialiserWidgets\(\)](#)

```
projet.lasalle84.darts.MainActivity.initialiserWidgets ( ) [private]
```

Récupère les widgets du layout de l'activité

Définition à la ligne 105 du fichier [MainActivity.java](#).

Référencé par [projet.lasalle84.darts.MainActivity.onCreate\(\)](#).

```
00106     {
00107         Log.d(TAG, "initialiserWidgets()");
00108
00109         // Rend les boutons cliquables
00110         boutonDemarrerPartie.setOnClickListener(this);
00111         boutonGererJoueurs.setOnClickListener(this);
00112         boutonVoirHistorique.setOnClickListener(this);
00113
00114         boutonGererJoueurs.setVisibility(View.INVISIBLE);
00115         boutonVoirHistorique.setVisibility(View.INVISIBLE);
00116     }
```

### 7.8.2.2 [onClick\(\)](#)

```
projet.lasalle84.darts.MainActivity.onClick (
    View element )
```

Méthode appelée lors d'un clic sur un widget de l'activité

## Paramètres

<i>element</i>	le widget cliqué
----------------	------------------

Définition à la ligne 63 du fichier [MainActivity.java](#).

```

00064     {
00065         Log.d(TAG, "onClick()");
00066
00067         if(element == boutonDemarrerPartie)
00068         {
00069             Log.d(TAG, "clic boutonDemarrerPartie");
00070             Intent activiteCreerPartie = new Intent(MainActivity.this, ActiviteCreerPartie.class);
00071             startActivity(activiteCreerPartie);
00072         }
00073     }
00074     else if(element == boutonGererJoueurs)
00075     {
00076         Log.d(TAG, "clic boutonGererJoueurs");
00077     }
00078     else if(element == boutonVoirHistorique)
00079     {
00080         Log.d(TAG, "clic boutonVoirHistorique");
00081     }
00082 }

```

## 7.8.2.3 onCreate()

```

projet.lasalle84.darts.MainActivity.onCreate (
    Bundle savedInstanceState ) [protected]

```

Méthode appelée à la création de l'activité

## Paramètres

<i>savedInstanceState</i>	
---------------------------	--

Définition à la ligne 43 du fichier [MainActivity.java](#).

Références [projet.lasalle84.darts.MainActivity.initialiserWidgets\(\)](#), et [projet.lasalle84.darts.MainActivity.recupererWidgets\(\)](#).

```

00044     {
00045         super.onCreate(savedInstanceState);
00046         Log.d(TAG, "onCreate()");
00047
00048         // Associer le layout à l'activité
00049         setContentView(R.layout.activity_main);
00050
00051         recupererWidgets();
00052
00053         initialiserWidgets();
00054     }

```

## 7.8.2.4 recupererWidgets()

```

projet.lasalle84.darts.MainActivity.recupererWidgets ( ) [private]

```

Récupère les widgets du layout de l'activité

Définition à la ligne 90 du fichier [MainActivity.java](#).

Référencé par [projet.lasalle84.darts.MainActivity.onCreate\(\)](#).

```
00091     {
00092         Log.d(TAG, "recupererWidgets()");
00093
00094         boutonDemarrerPartie = (Button) findViewById(R.id.boutonDemarrerPartie);
00095         boutonGererJoueurs = (Button) findViewById(R.id.boutonGererJoueurs);
00096         boutonVoirHistorique = (Button) findViewById(R.id.boutonVoirHistorique);
00097     }
```

### 7.8.3 Documentation des données membres

#### 7.8.3.1 boutonDemarrerPartie

```
Button projet.lasalle84.darts.MainActivity.boutonDemarrerPartie [private]
```

Widgets

Définition à la ligne 31 du fichier [MainActivity.java](#).

#### 7.8.3.2 boutonGererJoueurs

```
Button projet.lasalle84.darts.MainActivity.boutonGererJoueurs [private]
```

Définition à la ligne 32 du fichier [MainActivity.java](#).

#### 7.8.3.3 boutonVoirHistorique

```
Button projet.lasalle84.darts.MainActivity.boutonVoirHistorique [private]
```

Définition à la ligne 33 du fichier [MainActivity.java](#).

#### 7.8.3.4 TAG

```
final String projet.lasalle84.darts.MainActivity.TAG = "IHMEcranPrincipal" [static], [private]
```

le TAG de la classe pour les logs

Constantes

Définition à la ligne 26 du fichier [MainActivity.java](#).

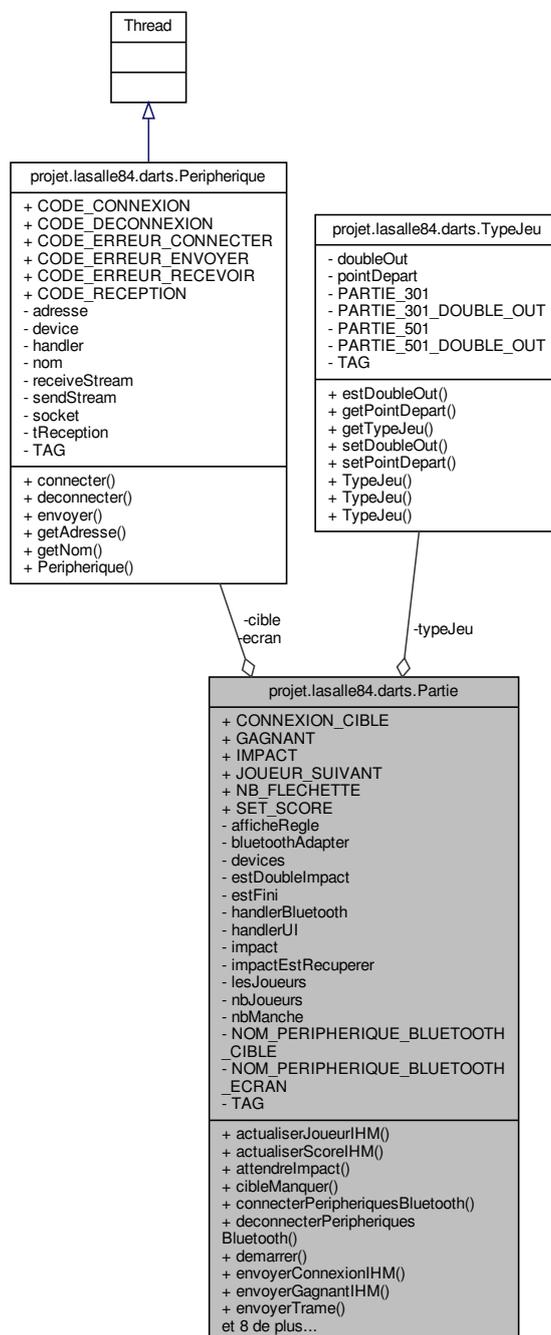
La documentation de cette classe a été générée à partir du fichier suivant :

— [MainActivity.java](#)

## 7.9 Référence de la classe projet.lasalle84.darts.Partie

Déclaration de la classe [Partie](#).

Grphe de collaboration de projet.lasalle84.darts.Partie :



## Fonctions membres publiques

- void `actualiserJoueurIHM` (String monJoueur)  
*Actualiser le joueur à IHM.*
- void `actualiserScoreIHM` (Joueur monJoueur, int score)

- *Actualiser le score du joueur à IHM.*
- void `attendreImpact ()`
- *Attendre l'impact.*
- void `cibleManquer ()`
- void `connecterPeripheriquesBluetooth ()`
- *Se connecter sur un périphérique via son adresse.*
- void `deconnecterPeripheriquesBluetooth ()`
- *Deconnecter les périphériques bluetooth.*
- void `demarrer ()`
- *Démarrer la partie.*
- void `envoyerConnexionIHM (String peripherique)`
- void `envoyerGagnantIHM (Joueur monJoueur)`
- *Envoyer le gagnant a IHM.*
- void `envoyerTrame (Peripherique peripherique, String trame)`
- *Envoie une trame à un périphérique.*
- `TypeJeu getTypeJeu ()`
- void `impactIHM (int typePoint, int numeroCible)`
- *Envoyer impact à IHM.*
- void `initialiserPartie ()`
- *Initialiser la partie.*
- `Partie (ArrayList< Joueur > lesJoueurs, TypeJeu typeJeu, boolean AfficheRegle, BluetoothAdapter bluetoothAdapter, Handler handlerUI)`
- *Constructeur de la classe Partie.*
- void `pause ()`
- void `recupererPeripheriques ()`
- *Récupère les périphériques Bluetooth qui sont appairés.*
- void `reprendre ()`
- void `traitementTrame (String trame)`
- *traiter la trame reçu par Handler*

#### Attributs publics statiques

- static final int `CONNEXION_CIBLE = 4`
- static final int `GAGNANT = 3`
- static final int `IMPACT = 2`
- static final int `JOUEUR_SUIVANT = 0`
- static final int `NB_FLECHETTE = 3`
- static final int `SET_SCORE = 1`

#### Attributs privés

- boolean `afficheRegle = true`
- *Booléen si on doit afficher les règles de la partie.*
- BluetoothAdapter `bluetoothAdapter`
- *Bluetooth Adapteur.*
- `Peripherique cible = null`
- *Peripherique darts connecté en Bluetooth.*
- Set< BluetoothDevice > `devices`
- *Les peripheriques qui sont appairés.*
- `Peripherique ecran = null`
- *Peripherique raspberry connecté en Bluetooth.*
- boolean `estDoubleImpact = false`
- *Booléen si la dernière impact est du type double.*
- boolean `estFini = false`
- *Booléen si la partie est fini.*
- Handler `handlerBluetooth`
- Handler `handlerUI = null`
- *Handler pour gérer l'interface.*
- int `impact [] = null`
- *Les impacts.*
- boolean `impactEstRecuperer`
- *Booléen pour savoir quand on reçoit une trame impact.*
- ArrayList< Joueur > `lesJoueurs`
- *Les objets Joueur stocker dans un conteneur (Queue)*
- int `nbJoueurs`
- *Nombre de joueur.*
- int `nbManche`
- *Nombre de manche.*
- `TypeJeu typeJeu`
- *Mode de jeu.*

## Attributs privés statiques

- static final String `NOM_PERIPHERIQUE_BLUETOOTH_CIBLE` = "impact-darts"  
*le nom du périphérique Bluetooth du module cible*
- static final String `NOM_PERIPHERIQUE_BLUETOOTH_ECRAN` = "ecran-darts"  
*le nom du périphérique Bluetooth du module écran*
- static final String `TAG` = "Partie"  
*Tag pour Log.*

## 7.9.1 Description détaillée

Déclaration de la classe [Partie](#).

Définition à la ligne 26 du fichier [Partie.java](#).

## 7.9.2 Documentation des constructeurs et destructeur

## 7.9.2.1 Partie()

```
projet.lasalle84.darts.Partie.Partie (
    ArrayList< Joueur > lesJoueurs,
    TypeJeu typeJeu,
    boolean AfficheRegle,
    BluetoothAdapter bluetoothAdapter,
    Handler handlerUI )
```

Constructeur de la classe [Partie](#).

## Paramètres

<i>lesJoueurs</i>	
<i>typeJeu</i>	

Définition à la ligne 68 du fichier [Partie.java](#).

Références [projet.lasalle84.darts.Partie.bluetoothAdapter](#), [projet.lasalle84.darts.Partie.connecterPeripheriquesBluetooth\(\)](#), [projet.lasalle84.darts.Partie.handlerUI](#), [projet.lasalle84.darts.Partie.lesJoueurs](#), [projet.lasalle84.darts.Partie.recupererPeripheriques\(\)](#), et [projet.lasalle84.darts.Partie.typeJeu](#).

```
00069     {
00070         this.lesJoueurs = lesJoueurs;
00071         this.typeJeu = typeJeu;
00072         this.bluetoothAdapter = bluetoothAdapter;
00073         this.handlerUI = handlerUI;
00074         this.nbManche = 0;
00075         this.nbJoueurs = this.lesJoueurs.size();
00076         impact = new int[2];
00077         this.afficheRegle = AfficheRegle;
00078         recupererPeripheriques();
00079         connecterPeripheriquesBluetooth();
00080     }
00081 }
```

## 7.9.3 Documentation des fonctions membres

### 7.9.3.1 actualiserJoueurIHM()

```
projet.lasalle84.darts.Partie.actualiserJoueurIHM (
    String monJoueur )
```

Actualiser le joueur à IHM.

Définition à la ligne 389 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

```
00390     {
00391         Log.d(TAG, "actualiserJoueurIHM()");
00392         Message msg = Message.obtain();
00393         Bundle b = new Bundle();
00394         b.putInt("action", JOUEUR_SUIVANT);
00395         b.putString("joueur", monJoueur);
00396         msg.setData(b);
00397         handlerUI.sendMessage(msg);
00398     }
```

### 7.9.3.2 actualiserScoreIHM()

```
projet.lasalle84.darts.Partie.actualiserScoreIHM (
    Joueur monJoueur,
    int score )
```

Actualiser le score du joueur à IHM.

Définition à la ligne 371 du fichier [Partie.java](#).

Références [projet.lasalle84.darts.Joueur.getNom\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#), et [projet.lasalle84.darts.Partie.initialiserPartie\(\)](#).

```
00372     {
00373         Log.d(TAG, "actualiserScoreIHM()");
00374         Message msg = Message.obtain();
00375         Bundle b = new Bundle();
00376         b.putInt("action", SET_SCORE);
00377         b.putString("joueur", monJoueur.getNom());
00378         b.putInt("score", score);
00379         msg.setData(b);
00380         handlerUI.sendMessage(msg);
00381     }
```

### 7.9.3.3 attendreImpact()

```
projet.lasalle84.darts.Partie.attendreImpact ( )
```

Attendre l'impact.

Définition à la ligne 300 du fichier [Partie.java](#).

Références [projet.lasalle84.darts.Partie.envoyerTrame\(\)](#), et [projet.lasalle84.darts.Partie.impactIHM\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

```
00300     {
00301         Log.d(TAG, "attendreImpact()");
00302         while (!impactEstRecuperer)
00303         {
00304             sleep(1000);
00305         }
00306         envoyerTrame(ecran,"$DART;GAME;" + impact[0] + ";" +
00307 impact[1] + "\r\n");
00308         impactIHM(impact[0], impact[1]);
00309
00310         if (impact[0] == 2)
00311         {
00312             estDoubleImpact = true;
00313         }
00314         else
00315         {
00316             estDoubleImpact = false;
00317         }
00318
00319         Log.d(TAG, "Type cible: " + impact[0] + "Numero Cible:" +
impact[1]);
00320     }
00321 }
```

### 7.9.3.4 cibleManquer()

```
void projet.lasalle84.darts.Partie.cibleManquer ( )
```

Définition à la ligne 436 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onClick\(\)](#).

```
00437     {
00438         impact[0] = 0;
00439         impact[1] = 0;
00440         impactEstRecuperer = true;
00441     }
```

### 7.9.3.5 connecterPeripheriquesBluetooth()

`projet.lasalle84.darts.Partie.connecterPeripheriquesBluetooth ( )`

Se connecter sur un périphérique via son adresse.

Définition à la ligne 100 du fichier `Partie.java`.

Références `projet.lasalle84.darts.Peripherique.connecter()`, `projet.lasalle84.darts.Partie.handlerBluetooth`, `projet.lasalle84.darts.Partie.NOM_PERIPHERIQUE_BLUETOOTH_CIBLE`, et `projet.lasalle84.darts.Partie.NOM_PERIPHERIQUE_BLUETOOTH_ECRAN`.

Référencé par `projet.lasalle84.darts.Partie.Partie()`.

```

00101     {
00102         Log.d(TAG, "connecterPeripheriquesBluetooth()");
00103
00104         String nomPeripherique = NOM_PERIPHERIQUE_BLUETOOTH_ECRAN;
00105         for (BluetoothDevice device : devices)
00106         {
00107             if(device.getName().contains(nomPeripherique))
00108             {
00109                 ecran = new Peripherique(device, handlerBluetooth);
00110                 Log.d(TAG, "Adresse du périphérique raspberrypi " + device.getAddress());
00111                 break;
00112             }
00113         }
00114
00115         nomPeripherique = NOM_PERIPHERIQUE_BLUETOOTH_CIBLE;
00116         for (BluetoothDevice device : devices)
00117         {
00118             if(device.getName().contains(nomPeripherique))
00119             {
00120                 cible = new Peripherique(device, handlerBluetooth);
00121                 Log.d(TAG, "Adresse du périphérique darts " + device.getAddress());
00122                 break;
00123             }
00124         }
00125
00126         if(ecran != null)
00127             ecran.connecter();
00128         if(cible != null)
00129             cible.connecter();
00130     }

```

### 7.9.3.6 deconnecterPeripheriquesBluetooth()

`projet.lasalle84.darts.Partie.deconnecterPeripheriquesBluetooth ( )`

Deconnecter les périphériques bluetooth.

Définition à la ligne 138 du fichier `Partie.java`.

Références `projet.lasalle84.darts.Peripherique.deconnecter()`.

Référencé par `projet.lasalle84.darts.Partie.envoyerGagnantIHM()`, et `projet.lasalle84.darts.ActivitePartie.finish()`.

```

00139     {
00140         Log.d(TAG, "deconnecterPeripheriquesBluetooth()");
00141         if (ecran != null)
00142         {
00143             ecran.deconnecter();
00144         }
00145
00146         if (cible != null)
00147         {
00148             cible.deconnecter();
00149         }
00150     }

```

## 7.9.3.7 demarrer()

```
projet.lasalle84.darts.Partie.demarrer ( )
```

Démarrer la partie.

Définition à la ligne 172 du fichier [Partie.java](#).

Références [projet.lasalle84.darts.Partie.actualiserJoueurIHM\(\)](#), [projet.lasalle84.darts.Partie.actualiserScoreIHM\(\)](#), [projet.lasalle84.darts.Partie.attendreImpact\(\)](#), [projet.lasalle84.darts.Partie.envoyerGagnantIHM\(\)](#), [projet.lasalle84.darts.Partie.estDoubleImpact](#), [projet.lasalle84.darts.TypeJeu.estDoubleOut\(\)](#), [projet.lasalle84.darts.Joueur.estEliminer](#), [projet.lasalle84.darts.Joueur.getNom\(\)](#), [projet.lasalle84.darts.Joueur.getScore\(\)](#), [projet.lasalle84.darts.Partie.initialiserPartie\(\)](#), [projet.lasalle84.darts.Partie.NB\\_FLECHE](#), et [projet.lasalle84.darts.Joueur.retirerPoint\(\)](#).

Référencé par [projet.lasalle84.darts.ActivitePartie.onClick\(\)](#).

```
00172         {
00173             Log.d(TAG, "demarrer()");
00174             initialiserPartie();
00175
00176             do
00177             {
00178                 nbManche++;
00179                 Iterator<Joueur> it = lesJoueurs.iterator();
00180                 Log.d(TAG, "Manche numéro " + nbManche);
00181                 while (it.hasNext())
00182                 {
00183                     Joueur monJoueur = it.next();
00184                     Log.d(TAG, "c'est le tour à " + monJoueur.getNom());
00185                     actualiserJoueurIHM(monJoueur.getNom());
00186                     int pointVolley = 0;
00187
00188                     for(int i = 0; i < NB_FLECHETTE; i++)
00189                     {
00190                         if (monJoueur.estEliminer())
00191                         {
00192                             i = NB_FLECHETTE;
00193                         }
00194                         else
00195                         {
00196                             impactEstRecuperer = false;
00197                             attendreImpact();
00198                             pointVolley += impact[0]*impact[1];
00199                             if(!monJoueur.retirerPoint(pointVolley, this))
00200                             {
00201                                 i = NB_FLECHETTE;
00202                                 pointVolley = 0;
00203                             }
00204                             else if (monJoueur.getScore() == 0 && !typeJeu.
estDoubleOut())
00205                             {
00206                                 envoyerGagnantIHM(monJoueur);
00207                                 i = NB_FLECHETTE;
00208                                 estFini = true;
00209                             }
00210                             else if (monJoueur.getScore() == 0 && typeJeu.
estDoubleOut() && estDoubleImpact)
00211                             {
00212                                 envoyerGagnantIHM(monJoueur);
00213                                 i = NB_FLECHETTE;
00214                                 estFini = true;
00215                             }
00216                         }
00217                     }
00218                 }
00219                 actualiserScoreIHM(monJoueur, monJoueur.getScore());
00220             }
00221             }while (!estFini);
00222         }
```

### 7.9.3.8 envoyerConnexionIHM()

```
void projet.lasalle84.darts.Partie.envoyerConnexionIHM (
    String peripherique )
```

Définition à la ligne 443 du fichier [Partie.java](#).

```
00444     {
00445         if (peripherique.equals(NOM_PERIPHERIQUE_BLUETOOTH_CIBLE))
00446         {
00447             Log.d(TAG, "envoyerConnexionIHM() ");
00448             Message msg = Message.obtain();
00449             Bundle b = new Bundle();
00450             b.putInt("action", CONNEXION_CIBLE);
00451             msg.setData(b);
00452             handlerUI.sendMessage(msg);
00453         }
00454     }
```

### 7.9.3.9 envoyerGagnantIHM()

```
projet.lasalle84.darts.Partie.envoyerGagnantIHM (
    Joueur monJoueur )
```

Envoyer le gagnant a IHM.

Définition à la ligne 424 du fichier [Partie.java](#).

Références [projet.lasalle84.darts.Partie.deconnecterPeripheriquesBluetooth\(\)](#), et [projet.lasalle84.darts.Joueur.getNom\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

```
00424     {
00425         Log.d(TAG, "envoyerGagnantIHM() ");
00426         estFini = true;
00427         deconnecterPeripheriquesBluetooth();
00428         Message msg = Message.obtain();
00429         Bundle b = new Bundle();
00430         b.putInt("action", GAGNANT);
00431         b.putString("gagnant", monJoueur.getNom());
00432         msg.setData(b);
00433         handlerUI.sendMessage(msg);
00434     }
```

### 7.9.3.10 envoyerTrame()

```
projet.lasalle84.darts.Partie.envoyerTrame (
    Peripherique peripherique,
    String trame )
```

Envoie une trame à un périphérique.

Définition à la ligne 158 du fichier [Partie.java](#).

Références [projet.lasalle84.darts.Peripherique.envoyer\(\)](#), et [projet.lasalle84.darts.Peripherique.getNom\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.attendreImpact\(\)](#), et [projet.lasalle84.darts.Partie.initialiserPartie\(\)](#).

```
00159     {
00160         if(peripherique != null)
00161         {
00162             Log.d(TAG, "envoyerTrame() " + peripherique.getNom());
00163             peripherique.envoyer(trame);
00164         }
00165     }
```

## 7.9.3.11 getTypeJeu()

TypeJeu projet.lasalle84.darts.Partie.getTypeJeu ( )

Définition à la ligne 466 du fichier Partie.java.

Références projet.lasalle84.darts.Partie.typeJeu.

Référencé par projet.lasalle84.darts.Joueur.retirerPoint().

```
00466         {
00467         return typeJeu;
00468     }
```

## 7.9.3.12 impactIHM()

```
projet.lasalle84.darts.Partie.impactIHM (
    int typePoint,
    int numeroCible )
```

Envoyer impact à IHM.

Définition à la ligne 406 du fichier Partie.java.

Référencé par projet.lasalle84.darts.Partie.attendreImpact().

```
00407     {
00408         Log.d(TAG, "impactIHM()");
00409         Message msg = Message.obtain();
00410         Bundle b = new Bundle();
00411         b.putInt("action", IMPACT);
00412         b.putInt("typePoint", typePoint);
00413         b.putInt("numeroCible", numeroCible);
00414         msg.setData(b);
00415         handlerUI.sendMessage(msg);
00416     }
```

## 7.9.3.13 initialiserPartie()

```
projet.lasalle84.darts.Partie.initialiserPartie ( )
```

Initialiser la partie.

Définition à la ligne 230 du fichier Partie.java.

Références projet.lasalle84.darts.Partie.actualiserScoreIHM(), projet.lasalle84.darts.Partie.envoyerTrame(), projet.lasalle84.darts.Joueur.getNom(), projet.lasalle84.darts.TypeJeu.getPointDepart(), projet.lasalle84.darts.TypeJeu.getTypeJeu(), et projet.lasalle84.darts.Joueur.setScore().

Référencé par projet.lasalle84.darts.Partie.demarrer().

```
00230         {
00231         Log.d(TAG, "initialiserPartie()");
00232         Iterator<Joueur> it = lesJoueurs.iterator();
00233         String nomJoueurTrame = new String();
00234         while(it.hasNext())
00235         {
00236             Joueur monJoueur = it.next();
00237             monJoueur.setScore(typeJeu.getPointDepart());
00238             actualiserScoreIHM(monJoueur, typeJeu.
getPointDepart());
00239             nomJoueurTrame = nomJoueurTrame.concat(monJoueur.getNom() + ";");
00240         }
00241         int afficheRegleInt = afficheRegle ? 1 : 0;
00242         envoyerTrame(ecran, "$DARTS;START;" + typeJeu.
getTypeJeu() + ";" + afficheRegleInt + ";" + lesJoueurs.size() + ";" + nomJoueurTrame +
"\r\n");
00243     }
```

#### 7.9.3.14 pause()

```
void projet.lasalle84.darts.Partie.pause ( )
```

Définition à la ligne [456](#) du fichier [Partie.java](#).

```
00457     {  
00458         //TODO pause()  
00459     }
```

#### 7.9.3.15 recupererPeripheriques()

```
projet.lasalle84.darts.Partie.recupererPeripheriques ( )
```

Récupère les périphériques Bluetooth qui sont appairés.

Définition à la ligne [89](#) du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.Partie\(\)](#).

```
00090     {  
00091         Log.d(TAG, "recupererPeripheriques()");  
00092         devices = bluetoothAdapter.getBondedDevices();  
00093     }
```

#### 7.9.3.16 reprendre()

```
void projet.lasalle84.darts.Partie.reprendre ( )
```

Définition à la ligne [461](#) du fichier [Partie.java](#).

```
00462     {  
00463         //TODO reprendre()  
00464     }
```

### 7.9.3.17 traitementTrame()

```
projet.lasalle84.darts.Partie.traitementTrame (
    String trame )
```

traiter la trame reçu par Handler

Définition à la ligne 329 du fichier [Partie.java](#).

```
00330     {
00331         trame = trame.replace("\r\n", "");
00332         String[] trameDecoupe = trame.split(";",4);
00333         Bundle bundle = new Bundle();
00334
00335         switch (trameDecoupe[1])
00336         {
00337             case "GAME":
00338                 Log.d(TAG, "traitementTrame: GAME");
00339                 impact[0] = Integer.parseInt(trameDecoupe[2]);
00340                 impact[1] = Integer.parseInt(trameDecoupe[3]);
00341                 impactEstRecuperer = true;
00342
00343                 break;
00344             case "PAUSE":
00345                 //TODO PAUSE
00346                 break;
00347             case "PLAY":
00348                 //TODO PLAY
00349                 break;
00350             case "STOP":
00351                 //TODO STOP
00352                 break;
00353             case "RESET":
00354                 //TODO RESET
00355                 break;
00356             case "HEARTBEAT ":
00357                 //TODO HEARTBEAT
00358                 break;
00359             case "ACK":
00360                 //TODO ACK
00361                 break;
00362         }
00363     }
```

## 7.9.4 Documentation des données membres

### 7.9.4.1 afficheRegle

```
boolean projet.lasalle84.darts.Partie.afficheRegle = true [private]
```

Booléen si on doit afficher les règles de la partie.

Définition à la ligne 59 du fichier [Partie.java](#).

### 7.9.4.2 bluetoothAdapter

```
BluetoothAdapter projet.lasalle84.darts.Partie.bluetoothAdapter [private]
```

Bluetooth Adapteur.

Définition à la ligne 50 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.Partie\(\)](#).

### 7.9.4.3 cible

```
Peripherique projet.lasalle84.darts.Partie.cible = null [private]
```

Peripherique darts connecté en Bluetooth.

Définition à la ligne 53 du fichier [Partie.java](#).

### 7.9.4.4 CONNEXION\_CIBLE

```
final int projet.lasalle84.darts.Partie.CONNEXION_CIBLE = 4 [static]
```

Définition à la ligne 42 du fichier [Partie.java](#).

### 7.9.4.5 devices

```
Set<BluetoothDevice> projet.lasalle84.darts.Partie.devices [private]
```

Les peripheriques qui sont appairés.

Définition à la ligne 51 du fichier [Partie.java](#).

### 7.9.4.6 ecran

```
Peripherique projet.lasalle84.darts.Partie.ecran = null [private]
```

Peripherique raspberry connecté en Bluetooth.

Définition à la ligne 52 du fichier [Partie.java](#).

### 7.9.4.7 estDoubleImpact

```
boolean projet.lasalle84.darts.Partie.estDoubleImpact = false [private]
```

Booléen si la derniere impact est du type double.

Définition à la ligne 58 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

### 7.9.4.8 estFini

```
boolean projet.lasalle84.darts.Partie.estFini = false [private]
```

Booléen si la partie est fini.

Définition à la ligne 57 du fichier [Partie.java](#).

#### 7.9.4.9 GAGNANT

```
final int projet.lasalle84.darts.Partie.GAGNANT = 3 [static]
```

Définition à la ligne 41 du fichier [Partie.java](#).

#### 7.9.4.10 handlerBluetooth

```
Handler projet.lasalle84.darts.Partie.handlerBluetooth [private]
```

Définition à la ligne 246 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.connecterPeripheriquesBluetooth\(\)](#).

#### 7.9.4.11 handlerUI

```
Handler projet.lasalle84.darts.Partie.handlerUI = null [private]
```

Handler pour gérer l'interface.

Définition à la ligne 54 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.Partie\(\)](#).

#### 7.9.4.12 IMPACT

```
final int projet.lasalle84.darts.Partie.IMPACT = 2 [static]
```

Définition à la ligne 40 du fichier [Partie.java](#).

#### 7.9.4.13 impact

```
int projet.lasalle84.darts.Partie.impact[] = null [private]
```

Les impacts.

Définition à la ligne 55 du fichier [Partie.java](#).

#### 7.9.4.14 impactEstRecuperer

```
boolean projet.lasalle84.darts.Partie.impactEstRecuperer [private]
```

Booléen pour savoir quand on reçoit une trame impact.

Définition à la ligne 56 du fichier [Partie.java](#).

#### 7.9.4.15 JOUEUR\_SUIVANT

```
final int projet.lasalle84.darts.Partie.JOUEUR_SUIVANT = 0 [static]
```

Code HandlerUI

Définition à la ligne 38 du fichier [Partie.java](#).

#### 7.9.4.16 lesJoueurs

```
ArrayList<Joueur> projet.lasalle84.darts.Partie.lesJoueurs [private]
```

Les objets [Joueur](#) stocker dans un conteneur (Queue)

Définition à la ligne 49 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.Partie\(\)](#).

#### 7.9.4.17 NB\_FLECHETTE

```
final int projet.lasalle84.darts.Partie.NB_FLECHETTE = 3 [static]
```

Définition à la ligne 34 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#).

#### 7.9.4.18 nbJoueurs

```
int projet.lasalle84.darts.Partie.nbJoueurs [private]
```

Nombre de joueur.

Attributs

Définition à la ligne 46 du fichier [Partie.java](#).

#### 7.9.4.19 nbManche

```
int projet.lasalle84.darts.Partie.nbManche [private]
```

Nombre de manche.

Définition à la ligne 47 du fichier [Partie.java](#).

#### 7.9.4.20 `NOM_PERIPHERIQUE_BLUETOOTH_CIBLE`

```
final String projet.lasalle84.darts.Partie.NOM_PERIPHERIQUE_BLUETOOTH_CIBLE = "impact-darts" [static], [private]
```

le nom du périphérique Bluetooth du module cible

Définition à la ligne 33 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.connecterPeripheriquesBluetooth\(\)](#).

#### 7.9.4.21 `NOM_PERIPHERIQUE_BLUETOOTH_ECRAN`

```
final String projet.lasalle84.darts.Partie.NOM_PERIPHERIQUE_BLUETOOTH_ECRAN = "ecran-darts" [static], [private]
```

le nom du périphérique Bluetooth du module écran

Définition à la ligne 32 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.connecterPeripheriquesBluetooth\(\)](#).

#### 7.9.4.22 `SET_SCORE`

```
final int projet.lasalle84.darts.Partie.SET_SCORE = 1 [static]
```

Définition à la ligne 39 du fichier [Partie.java](#).

#### 7.9.4.23 `TAG`

```
final String projet.lasalle84.darts.Partie.TAG = "Partie" [static], [private]
```

Tag pour Log.

Constantes

Définition à la ligne 31 du fichier [Partie.java](#).

#### 7.9.4.24 `typeJeu`

```
TypeJeu projet.lasalle84.darts.Partie.typeJeu [private]
```

Mode de jeu.

Définition à la ligne 48 du fichier [Partie.java](#).

Référencé par [projet.lasalle84.darts.Partie.getTypeJeu\(\)](#), et [projet.lasalle84.darts.Partie.Partie\(\)](#).

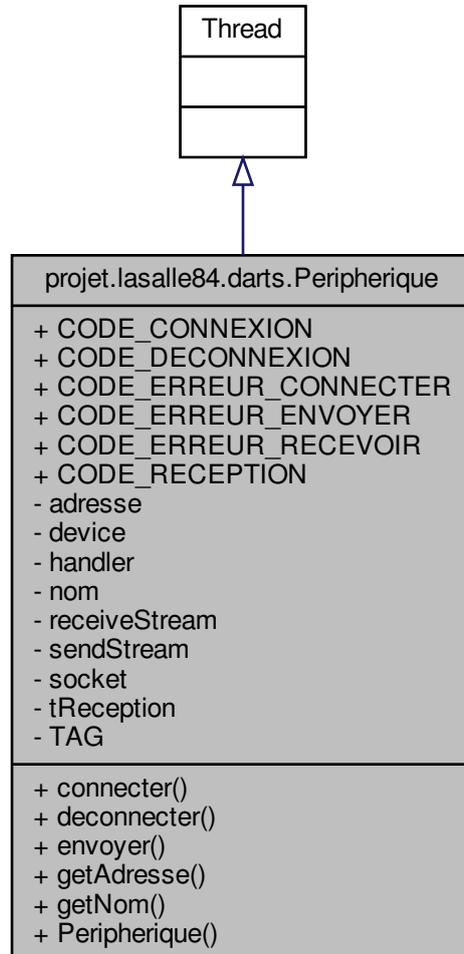
La documentation de cette classe a été générée à partir du fichier suivant :

— [Partie.java](#)

## 7.10 Référence de la classe projet.lasalle84.darts.Peripherique

Déclaration de la classe [Peripherique](#).

Graphe de collaboration de projet.lasalle84.darts.Peripherique :



### Fonctions membres publiques

- void `connecter` ()  
*Méthode pour se connecter sur le perihperique en Bluetooth.*
- boolean `deconnecter` ()  
*Méthode pour se deconnecter sur le perihperique en Bluetooth.*
- void `envoyer` (final String data)  
*Méthode pour envoyer une trame en Bluetooth.*
- String `getAdresse` ()  
*Méthode qui retourne l'adresse du périphérique.*
- String `getNom` ()  
*Méthode qui retourne le nom du périphérique.*
- `Peripherique` (BluetoothDevice `device`, Handler `handler`)  
*Constructeur de la classe `Peripherique`.*

### Attributs publics statiques

- static final int `CODE_CONNEXION` = 0  
*Code de Connection.*
- static final int `CODE_DECONNEXION` = 2  
*Code de Deconnexion.*
- static final int `CODE_ERREUR_CONNECTER` = -3  
*Code erreur lors de la connexion.*
- static final int `CODE_ERREUR_ENVOYER` = -1  
*Code erreur lors de l'envoi.*
- static final int `CODE_ERREUR_RECEVOIR` = -2  
*Code erreur lors de la réception.*
- static final int `CODE_RECEPTION` = 1  
*Code de Reception.*

### Attributs privés

- String `adresse`  
*Adresse du peripherique bluetooth.*
- BluetoothDevice `device`  
*Objet BluetoothDevice.*
- Handler `handler`  
*Handler permet de traiter les trames.*
- String `nom`  
*Nom du peripherique bluetooth.*
- InputStream `receiveStream` = null  
*Input du Bluetooth.*
- OutputStream `sendStream` = null  
*Output du Bluetooth.*
- BluetoothSocket `socket` = null  
*Socket Bluetooth.*
- Thread `tReception`  
*Thread pour traiter les trames.*

### Attributs privés statiques

- static final String `TAG` = "Peripherique"  
*TAG pour log.*

#### 7.10.1 Description détaillée

Déclaration de la classe `Peripherique`.

Définition à la ligne 26 du fichier `Peripherique.java`.

#### 7.10.2 Documentation des constructeurs et destructeur

##### 7.10.2.1 `Peripherique()`

```
projet.lasalle84.darts.Peripherique.Peripherique (
    BluetoothDevice device,
    Handler handler )
```

Constructeur de la classe `Peripherique`.

## Paramètres

<code>device,handler</code>
-----------------------------

Définition à la ligne 55 du fichier `Peripherique.java`.

Références `projet.lasalle84.darts.Peripherique.device`, et `projet.lasalle84.darts.Peripherique.handler`.

```

00056     {
00057         Log.d(TAG,"Peripherique() " + device.getName() + "[" + device.getAddress() + "]");
00058         if(device != null)
00059         {
00060             this.device = device;
00061             this.nom = device.getName();
00062             this.adresse = device.getAddress();
00063             this.handler = handler;
00064         }
00065         else
00066         {
00067             this.device = device;
00068             this.nom = "Aucun";
00069             this.adresse = "";
00070             this.handler = handler;
00071         }
00072
00073         try
00074         {
00075             assert device != null;
00076             socket = device.createRfcommSocketToServiceRecord(UUID.fromString("
00077 00001101-0000-1000-8000-00805F9B34FB"));
00078             receiveStream = socket.getInputStream();
00079             sendStream = socket.getOutputStream();
00080         }
00081         catch (IOException e)
00082         {
00083             e.printStackTrace();
00084             socket = null;
00085         }
00086         if(socket != null)
00087             tReception = new TReception(this, handler,
00088             receiveStream);
00089     }

```

### 7.10.3 Documentation des fonctions membres

#### 7.10.3.1 connecter()

`projet.lasalle84.darts.Peripherique.connecter ( )`

Méthode pour se connecter sur le perihperique en Bluetooth.

Définition à la ligne 114 du fichier `Peripherique.java`.

Références `projet.lasalle84.darts.Peripherique.getAdresse()`, et `projet.lasalle84.darts.Peripherique.getNom()`.

Référencé par `projet.lasalle84.darts.Partie.connecterPeripheriquesBluetooth()`.

```

00115     {
00116         Log.d(TAG,"connecter() " + device.getName() + "[" + device.getAddress() + "]");
00117         new Thread()
00118         {
00119             @Override public void run()
00120             {
00121                 try
00122                 {
00123                     socket.connect();
00124
00125                     Message msg = Message.obtain();

```

```

00126         Bundle b = new Bundle();
00127         b.putString("nom", getNom());
00128         b.putString("adresse", getAdresse());
00129         b.putInt("etat", CODE_CONNEXION);
00130         msg.setData(b);
00131         handler.sendMessage(msg);
00132
00133         // on démarre le thread de réception
00134         tReception.start();
00135         Log.d(TAG, "connexion reussie " + getNom());
00136
00137     }
00138     catch (IOException e)
00139     {
00140         e.printStackTrace();
00141         Log.d(TAG, "erreur connexion " + getNom());
00142         Message msg = Message.obtain();
00143         Bundle b = new Bundle();
00144         b.putString("nom", getNom());
00145         b.putString("adresse", getAdresse());
00146         b.putInt("etat", CODE_ERREUR_CONNECTER);
00147         msg.setData(b);
00148         handler.sendMessage(msg);
00149     }
00150 }
00151 }.start();
00152 }

```

### 7.10.3.2 deconnecter()

```
projet.lasalle84.darts.Peripherique.deconnecter ( )
```

Méthode pour se deconnecter sur le perihperique en Bluetooth.

Définition à la ligne 158 du fichier [Peripherique.java](#).

Référencé par [projet.lasalle84.darts.Partie.deconnecterPeripheriquesBluetooth\(\)](#).

```

00159 {
00160     Log.d(TAG, "deconnecter() " + device.getName() + "[" + device.getAddress() + "]");
00161     try
00162     {
00163         tReception.arreter();
00164
00165         socket.close();
00166         return true;
00167     }
00168     catch (IOException e)
00169     {
00170         System.out.println("<Socket> error close");
00171         e.printStackTrace();
00172         return false;
00173     }
00174 }

```

### 7.10.3.3 envoyer()

```
projet.lasalle84.darts.Peripherique.envoyer (
    final String data )
```

Méthode pour envoyer une trame en Bluetooth.

Définition à la ligne 180 du fichier [Peripherique.java](#).

Références [projet.lasalle84.darts.Peripherique.getAdresse\(\)](#), et [projet.lasalle84.darts.Peripherique.getNom\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.envoyerTrame\(\)](#).

```

00181     {
00182         Log.d(TAG, "envoyer() " + device.getName() + "[" + device.getAddress() + "]");
00183         if(socket == null)
00184             {
00185                 Log.d(TAG, "pas d'envoi");
00186                 return;
00187             }
00188         new Thread()
00189         {
00190             @Override public void run()
00191             {
00192                 try
00193                 {
00194                     if(socket.isConnected())
00195                     {
00196                         sendStream.write(data.getBytes());
00197                         sendStream.flush();
00198                         Log.d(TAG, "envoyer() trame envoyée : " + data);
00199                     }
00200                 }
00201                 catch (IOException e)
00202                 {
00203                     e.printStackTrace();
00204                     Log.d(TAG, "envoyer() Erreur socket write : " + getNom());
00205                     Message msg = Message.obtain();
00206                     Bundle b = new Bundle();
00207                     b.putString("nom", getNom());
00208                     b.putString("adresse", getAdresse());
00209                     b.putInt("etat", CODE_ERREUR_ENVOYER);
00210                     msg.setData(b);
00211                     handler.sendMessage(msg);
00212                 }
00213             }
00214         }.start();
00215     }

```

#### 7.10.3.4 getAdresse()

```
projet.lasalle84.darts.Peripherique.getAdresse ( )
```

Méthode qui retourne l'adresse du périphérique.

##### Renvoie

adresse l'adresse du périphérique

Définition à la ligne 105 du fichier [Peripherique.java](#).

Références [projet.lasalle84.darts.Peripherique.adresse](#).

Référencé par [projet.lasalle84.darts.Peripherique.connecter\(\)](#), et [projet.lasalle84.darts.Peripherique.envoyer\(\)](#).

```

00106     {
00107         return adresse;
00108     }

```

#### 7.10.3.5 getNom()

```
projet.lasalle84.darts.Peripherique.getNom ( )
```

Méthode qui retourne le nom du périphérique.

##### Renvoie

nom le nom du périphérique

Définition à la ligne 95 du fichier [Peripherique.java](#).

Références [projet.lasalle84.darts.Peripherique.nom](#).

Référencé par [projet.lasalle84.darts.Peripherique.connecter\(\)](#), [projet.lasalle84.darts.Peripherique.envoyer\(\)](#), et [projet.lasalle84.darts.Partie.envoyerTrame\(\)](#).

```

00096     {
00097         return nom;
00098     }

```

#### 7.10.4 Documentation des données membres

##### 7.10.4.1 adresse

```
String projet.lasalle84.darts.Peripherique.adresse [private]
```

Adresse du peripherique bluetooth.

Définition à la ligne 43 du fichier [Peripherique.java](#).

Référencé par [projet.lasalle84.darts.Peripherique.getAdresse\(\)](#).

##### 7.10.4.2 CODE\_CONNEXION

```
final int projet.lasalle84.darts.Peripherique.CODE_CONNEXION = 0 [static]
```

Code de Connection.

Définition à la ligne 32 du fichier [Peripherique.java](#).

##### 7.10.4.3 CODE\_DECONNEXION

```
final int projet.lasalle84.darts.Peripherique.CODE_DECONNEXION = 2 [static]
```

Code de Deconnexion.

Définition à la ligne 34 du fichier [Peripherique.java](#).

##### 7.10.4.4 CODE\_ERREUR\_CONNECTER

```
final int projet.lasalle84.darts.Peripherique.CODE_ERREUR_CONNECTER = -3 [static]
```

Code erreur lors de la connexion.

Définition à la ligne 37 du fichier [Peripherique.java](#).

##### 7.10.4.5 CODE\_ERREUR\_ENVOYER

```
final int projet.lasalle84.darts.Peripherique.CODE_ERREUR_ENVOYER = -1 [static]
```

Code erreur lors de l'envoi.

Définition à la ligne 35 du fichier [Peripherique.java](#).

#### 7.10.4.6 CODE\_ERREUR\_RECEVOIR

```
final int projet.lasalle84.darts.Peripherique.CODE_ERREUR_RECEVOIR = -2 [static]
```

Code erreur lors de la réception.

Définition à la ligne 36 du fichier [Peripherique.java](#).

#### 7.10.4.7 CODE\_RECEPTION

```
final int projet.lasalle84.darts.Peripherique.CODE_RECEPTION = 1 [static]
```

Code de Reception.

Définition à la ligne 33 du fichier [Peripherique.java](#).

#### 7.10.4.8 device

```
BluetoothDevice projet.lasalle84.darts.Peripherique.device [private]
```

Objet BluetoothDevice.

Attributs

Définition à la ligne 41 du fichier [Peripherique.java](#).

Référencé par [projet.lasalle84.darts.Peripherique.Peripherique\(\)](#).

#### 7.10.4.9 handler

```
Handler projet.lasalle84.darts.Peripherique.handler [private]
```

Handler permet de traiter les trames.

Définition à la ligne 44 du fichier [Peripherique.java](#).

Référencé par [projet.lasalle84.darts.Peripherique.Peripherique\(\)](#).

#### 7.10.4.10 nom

```
String projet.lasalle84.darts.Peripherique.nom [private]
```

Nom du peripherique bluetooth.

Définition à la ligne 42 du fichier [Peripherique.java](#).

Référencé par [projet.lasalle84.darts.Peripherique.getNom\(\)](#).

#### 7.10.4.11 receiveStream

```
InputStream projet.lasalle84.darts.Peripherique.receiveStream = null [private]
```

Input du Bluetooth.

Définition à la ligne 46 du fichier [Peripherique.java](#).

#### 7.10.4.12 sendStream

```
OutputStream projet.lasalle84.darts.Peripherique.sendStream = null [private]
```

Output du Bluetooth.

Définition à la ligne 47 du fichier [Peripherique.java](#).

#### 7.10.4.13 socket

```
BluetoothSocket projet.lasalle84.darts.Peripherique.socket = null [private]
```

Socket Bluetooth.

Définition à la ligne 45 du fichier [Peripherique.java](#).

#### 7.10.4.14 TAG

```
final String projet.lasalle84.darts.Peripherique.TAG = "Peripherique" [static], [private]
```

TAG pour log.

Constantes

Définition à la ligne 31 du fichier [Peripherique.java](#).

#### 7.10.4.15 tReception

```
TReception projet.lasalle84.darts.Peripherique.tReception [private]
```

[Thread](#) pour traiter les trames.

Définition à la ligne 48 du fichier [Peripherique.java](#).

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

— [Peripherique.java](#)

## 7.11 Référence de la classe Thread

Graphe de collaboration de Thread :



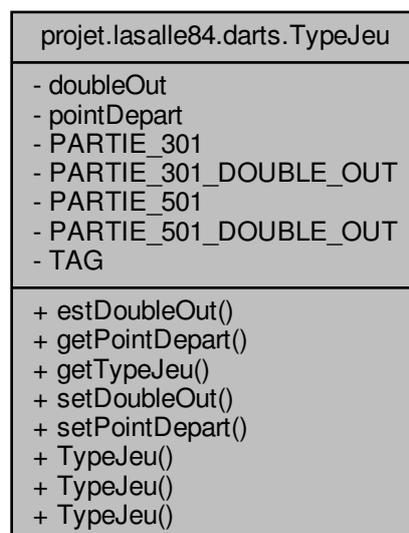
La documentation de cette classe a été générée à partir du fichier suivant :

— [TReception.java](#)

## 7.12 Référence de la classe projet.lasalle84.darts.TypeJeu

Déclaration de la classe [TypeJeu](#).

Graphe de collaboration de projet.lasalle84.darts.TypeJeu :



## Fonctions membres publiques

- boolean `estDoubleOut` ()
- int `getPointDepart` ()
- String `getTypeJeu` ()
- void `setDoubleOut` (boolean `doubleOut`)
- void `setPointDepart` (int `pointDepart`)
- `TypeJeu` ()  
*Constructeur par défaut de la classe `TypeJeu`.*
- `TypeJeu` (int `pointDepart`, boolean `doubleOut`)  
*Constructeur de la classe `TypeJeu`.*
- `TypeJeu` (int `idModeJeu`)  
*Constructeur de la classe `TypeJeu`.*

## Attributs privés

- boolean `doubleOut`  
*Le mode est en Double Out ?*
- int `pointDepart`  
*Point de depart du mode de jeu.*

## Attributs privés statiques

- static final int `PARTIE_301` = 2
- static final int `PARTIE_301_DOUBLE_OUT` = 3
- static final int `PARTIE_501` = 0
- static final int `PARTIE_501_DOUBLE_OUT` = 1
- static final String `TAG` = "TypeJeu"

## 7.12.1 Description détaillée

Déclaration de la classe `TypeJeu`.

Définition à la ligne 15 du fichier `TypeJeu.java`.

## 7.12.2 Documentation des constructeurs et destructeur

7.12.2.1 `TypeJeu()` [1/3]

```
projet.lasalle84.darts.TypeJeu.TypeJeu ( )
```

Constructeur par défaut de la classe `TypeJeu`.

Définition à la ligne 38 du fichier `TypeJeu.java`.

```
00039     {
00040         this.pointDepart = 501;
00041         this.doubleOut = true;
00042     }
```

7.12.2.2 `TypeJeu()` [2/3]

```
projet.lasalle84.darts.TypeJeu.TypeJeu (
    int pointDepart,
    boolean doubleOut )
```

Constructeur de la classe `TypeJeu`.

## Paramètres

<i>pointDepart</i>	
<i>doubleOut</i>	

Définition à la ligne 51 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.doubleOut](#), et [projet.lasalle84.darts.TypeJeu.pointDepart](#).

```
00052     {
00053         this.pointDepart = pointDepart;
00054         this.doubleOut = doubleOut;
00055     }
```

## 7.12.2.3 TypeJeu() [3/3]

```
projet.lasalle84.darts.TypeJeu.TypeJeu (
    int idModeJeu )
```

Constructeur de la classe [TypeJeu](#).

## Paramètres

<i>idModeJeu</i>	l'identifiant d'un type de jeu
------------------	--------------------------------

Définition à la ligne 63 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.PARTIE\\_301](#), [projet.lasalle84.darts.TypeJeu.PARTIE\\_301\\_DOUBLE\\_OUT](#), [projet.lasalle84.darts.TypeJeu.PARTIE\\_501](#), et [projet.lasalle84.darts.TypeJeu.PARTIE\\_501\\_DOUBLE\\_OUT](#).

```
00064     {
00065         switch (idModeJeu)
00066         {
00067             case PARTIE_501:
00068                 this.pointDepart = 501;
00069                 this.doubleOut = false;
00070                 break;
00071             case PARTIE_501_DOUBLE_OUT:
00072                 this.pointDepart = 501;
00073                 this.doubleOut = true;
00074                 break;
00075             case PARTIE_301:
00076                 this.pointDepart = 301;
00077                 this.doubleOut = false;
00078                 break;
00079             case PARTIE_301_DOUBLE_OUT:
00080                 this.pointDepart = 301;
00081                 this.doubleOut = true;
00082                 break;
00083         }
00084     }
00085 }
```

## 7.12.3 Documentation des fonctions membres

### 7.12.3.1 estDoubleOut()

```
boolean projet.lasalle84.darts.TypeJeu.estDoubleOut ( )
```

Définition à la ligne 96 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.doubleOut](#).

Référencé par [projet.lasalle84.darts.Partie.demarrer\(\)](#), et [projet.lasalle84.darts.Joueur.retirerPoint\(\)](#).

```
00097     {
00098         return doubleOut;
00099     }
```

### 7.12.3.2 getPointDepart()

```
int projet.lasalle84.darts.TypeJeu.getPointDepart ( )
```

Définition à la ligne 86 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.pointDepart](#).

Référencé par [projet.lasalle84.darts.TypeJeu.getTypeJeu\(\)](#), et [projet.lasalle84.darts.Partie.initialiserPartie\(\)](#).

```
00087     {
00088         return pointDepart;
00089     }
```

### 7.12.3.3 getTypeJeu()

```
String projet.lasalle84.darts.TypeJeu.getTypeJeu ( )
```

Définition à la ligne 106 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.getPointDepart\(\)](#).

Référencé par [projet.lasalle84.darts.Partie.initialiserPartie\(\)](#).

```
00107     {
00108         String typeJeu = Integer.toString(this.getPointDepart());
00109         if (doubleOut)
00110         {
00111             typeJeu = typeJeu.concat("_DOUBLE_OUT");
00112         }
00113         Log.d(TAG, "getTypeJeu: "+ typeJeu);
00114         return typeJeu;
00115     }
00116 }
```

#### 7.12.3.4 setDoubleOut()

```
void projet.lasalle84.darts.TypeJeu.setDoubleOut (
    boolean doubleOut )
```

Définition à la ligne 101 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.doubleOut](#).

```
00102     {
00103         this.doubleOut = doubleOut;
00104     }
```

#### 7.12.3.5 setPointDepart()

```
void projet.lasalle84.darts.TypeJeu.setPointDepart (
    int pointDepart )
```

Définition à la ligne 91 du fichier [TypeJeu.java](#).

Références [projet.lasalle84.darts.TypeJeu.pointDepart](#).

```
00092     {
00093         this.pointDepart = pointDepart;
00094     }
```

### 7.12.4 Documentation des données membres

#### 7.12.4.1 doubleOut

```
boolean projet.lasalle84.darts.TypeJeu.doubleOut [private]
```

Le mode est en Double Out ?

Attributs

Définition à la ligne 20 du fichier [TypeJeu.java](#).

Référencé par [projet.lasalle84.darts.TypeJeu.estDoubleOut\(\)](#), [projet.lasalle84.darts.TypeJeu.setDoubleOut\(\)](#), et [projet.lasalle84.darts.TypeJeu.TypeJeu\(\)](#).

#### 7.12.4.2 PARTIE\_301

```
final int projet.lasalle84.darts.TypeJeu.PARTIE_301 = 2 [static], [private]
```

Définition à la ligne 28 du fichier [TypeJeu.java](#).

Référencé par [projet.lasalle84.darts.TypeJeu.TypeJeu\(\)](#).

#### 7.12.4.3 PARTIE\_301\_DOUBLE\_OUT

```
final int projet.lasalle84.darts.TypeJeu.PARTIE_301_DOUBLE_OUT = 3 [static], [private]
```

Définition à la ligne 29 du fichier [TypeJeu.java](#).

Référencé par [projet.lasalle84.darts.TypeJeu.TypeJeu\(\)](#).

#### 7.12.4.4 PARTIE\_501

```
final int projet.lasalle84.darts.TypeJeu.PARTIE_501 = 0 [static], [private]
```

Constantes

Définition à la ligne 26 du fichier [TypeJeu.java](#).

Référencé par [projet.lasalle84.darts.TypeJeu.TypeJeu\(\)](#).

#### 7.12.4.5 PARTIE\_501\_DOUBLE\_OUT

```
final int projet.lasalle84.darts.TypeJeu.PARTIE_501_DOUBLE_OUT = 1 [static], [private]
```

Définition à la ligne 27 du fichier [TypeJeu.java](#).

Référencé par [projet.lasalle84.darts.TypeJeu.TypeJeu\(\)](#).

#### 7.12.4.6 pointDepart

```
int projet.lasalle84.darts.TypeJeu.pointDepart [private]
```

Point de depart du mode de jeu.

Définition à la ligne 21 du fichier [TypeJeu.java](#).

Référencé par [projet.lasalle84.darts.TypeJeu.getPointDepart\(\)](#), [projet.lasalle84.darts.TypeJeu.setPointDepart\(\)](#), et [projet.lasalle84.darts.TypeJeu.TypeJeu\(\)](#).

#### 7.12.4.7 TAG

```
final String projet.lasalle84.darts.TypeJeu.TAG = "TypeJeu" [static], [private]
```

Définition à la ligne 30 du fichier [TypeJeu.java](#).

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

— [TypeJeu.java](#)

## 8 Documentation des fichiers

### 8.1 Référence du fichier `ActiviteAjouterJoueur.java`

Déclaration de la classe `ActiviteAjouterJoueur`.

#### Classes

- class `projet.lasalle84.darts.ActiviteAjouterJoueur`  
*Déclaration de la classe `ActiviteAjouterJoueur`.*

#### Paquetages

- package `projet.lasalle84.darts`

#### 8.1.1 Description détaillée

Déclaration de la classe `ActiviteAjouterJoueur`.

Déclaration de la classe `ActivitePartie`.

#### Auteur

Menella Erwan

Définition dans le fichier `ActiviteAjouterJoueur.java`.

### 8.2 `ActiviteAjouterJoueur.java`

```

00001 package projet.lasalle84.darts;
00007 import androidx.appcompat.app.AppCompatActivity;
00008
00009 import android.app.Activity;
00010 import android.content.Intent;
00011 import android.os.Bundle;
00012 import android.util.Log;
00013 import android.view.View;
00014 import android.widget.Button;
00015 import android.widget.EditText;
00016
00017 import java.io.Serializable;
00018
00023 public class ActiviteAjouterJoueur extends AppCompatActivity implements View.
    OnClickListener {
00024
00025     private static final String TAG = "IHMAjouterJoueur";
00026     private Button boutonValider;
00027     private EditText saisir;
00028
00034     @Override
00035     protected void onCreate(Bundle savedInstanceState) {
00036         Log.d(TAG, "onCreate() ");
00037         super.onCreate(savedInstanceState);
00038         setContentView(R.layout.activity_ajouter_joueur);
00039         recupererWidget();
00040         initiliserWidget();
00041     }
00042
00049     public void recupererWidget()
00050     {
00051         Log.d(TAG, "recupererWidget() ");
00052         boutonValider = (Button) findViewById(R.id.bouton);
00053         saisir = (EditText) findViewById(R.id.Nom);
00054     }
00055
00062     public void initiliserWidget()

```

```
00063     {
00064         Log.d(TAG, "initiliserWidget () ");
00065         boutonValider.setOnClickListener(this);
00066         saisir.setOnClickListener(this);
00067     }
00068 }
00069
00076 @Override
00077 public void onClick(View element)
00078 {
00079     Log.d(TAG, "OnClick () ");
00080
00081     if(element == boutonValider)
00082     {
00083         Log.d(TAG, "clic BoutonValider ");
00084         finish();
00085     }
00086 }
00087
00093 @Override
00094 public void finish() {
00095     Log.d(TAG, "finish () ");
00096
00097     if (saisir.getText () != null)
00098     {
00099         Intent retour = new Intent();
00100         retour.putExtra("Joueur", saisir.getText ().toString().trim());
00101         setResult (RESULT_OK, retour);
00102         super.finish();
00103     }
00104 }
00105 }
```

### 8.3 Référence du fichier `ActiviteCreerPartie.java`

Déclaration de la classe `ActivitéCréerPartie`.

#### Classes

— class [projet.lasalle84.darts.ActiviteCreerPartie](#)

#### Paquetages

— package [projet.lasalle84.darts](#)

#### 8.3.1 Description détaillée

Déclaration de la classe `ActivitéCréerPartie`.

#### Auteur

Menella Erwan

Définition dans le fichier [ActiviteCreerPartie.java](#).

## 8.4 ActiviteCreerPartie.java

```

00001 package projet.lasalle84.darts;
00007 import androidx.appcompat.app.AppCompatActivity;
00008
00009 import android.content.Intent;
00010 import android.os.Bundle;
00011 import android.util.Log;
00012 import android.view.View;
00013 import android.widget.AdapterView;
00014 import android.widget.Button;
00015 import android.widget.CheckBox;
00016 import android.widget.ListView;
00017 import android.widget.Spinner;
00018
00019 import java.io.Serializable;
00020 import java.util.Vector;
00021
00022 public class ActiviteCreerPartie extends AppCompatActivity implements View.
    OnClickListener
00023 {
00024
00025     private static final String TAG = "IHMCreerPartie";
00026
00027     private Button boutonAjouterJoueur;
00028     private Spinner modeDeJeu;
00029     private ListView listViewJoueur;
00030     private ArrayAdapter<String> Adaptateur;
00031     private Button boutonLancerPartie;
00032     private CheckBox checkBoxAfficheRegle;
00033
00034     @Override
00035     protected void onCreate(Bundle savedInstanceState) {
00036         Log.d(TAG, "onCreate()");
00037         super.onCreate(savedInstanceState);
00038         setContentView(R.layout.activity_creer_partie);
00039         recupererWidgets();
00040         initialiserWidgets();
00041     }
00042
00043     @Override
00044     public void onClick(View element)
00045     {
00046         Log.d(TAG, "onClick()");
00047
00048         if(element == boutonAjouterJoueur)
00049         {
00050             Log.d(TAG, "clik boutonAjouterJoueur");
00051             Intent intent = new Intent(ActiviteCreerPartie.this,
00052                 ActiviteAjouterJoueur.class);
00053             startActivityForResult(intent,1);
00054         }
00055         else if (element == boutonLancerPartie)
00056         {
00057             Log.d(TAG,"clik boutonLancerPartie");
00058             LancerPartie();
00059         }
00060     }
00061
00062     public void LancerPartie()
00063     {
00064         Log.d(TAG, "LancerPartie()");
00065         Vector<Joueur>mesJoueurs = new Vector<>();
00066         if(Adaptateur.getCount() == 0)
00067         {
00068             for (int i = 0; i < 4; i++)
00069             {
00070                 mesJoueurs.add(new Joueur("Joueur" + i));
00071                 Log.d(TAG, "Joueur" + i + " = " + "Joueur" + i);
00072             }
00073         }
00074         else
00075         {
00076             for (int i = 0; i < Adaptateur.getCount(); i++)
00077             {
00078                 mesJoueurs.add(new Joueur(Adaptateur.getItem(i)));
00079                 Log.d(TAG, "Joueur" + i + " = " + Adaptateur.getItem(i));
00080             }
00081         }
00082
00083         Intent intent = new Intent(ActiviteCreerPartie.this,
00084             ActivitePartie.class);
00085         intent.putExtra("TypeMode",modeDeJeu.getSelectedItemPosition());
00086         Log.d(TAG, "TypeMode = " + modeDeJeu.getSelectedItemPosition());
00087         intent.putExtra("LesJoueurs", mesJoueurs);
00088         intent.putExtra("AfficheRegle", checkBoxAfficheRegle.isChecked());
00089         startActivity(intent);
00090         super.finish();
00091     }

```

```

00107     }
00108
00113     private void recupererWidgets()
00114     {
00115         Log.d(TAG, "recupererWidgets()");
00116
00117         boutonAjouterJoueur = (Button) findViewById(R.id.AjouterJoueur);
00118         modeDeJeu = (Spinner) findViewById(R.id.modeJeu);
00119         listViewJoueur = (ListView) findViewById(R.id.ListeJoueur);
00120         boutonLancerPartie = (Button) findViewById(R.id.boutonLancerPartie);
00121         checkBoxAfficheRegle = (CheckBox) findViewById(R.id.afficheRegle);
00122
00123     }
00124 }
00125
00130     private void initialiserWidgets()
00131     {
00132         Log.d(TAG, "initialiserWidgets()");
00133         boutonAjouterJoueur.setOnClickListener(this);
00134         boutonLancerPartie.setOnClickListener(this);
00135         Adaptateur = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
00136         listViewJoueur.setAdapter(Adaptateur);
00137     }
00138
00143     @Override
00144     public void onActivityResult(int requestCode, int resultCode, Intent data) {
00145         super.onActivityResult(requestCode, resultCode, data);
00146         if (requestCode == 1)
00147         {
00148             if (resultCode == RESULT_OK) {
00149                 String nomJoueur = data.getStringExtra("Joueur");
00150                 Log.d(TAG, "nom du joueur: " + nomJoueur);
00151                 Adaptateur.add(nomJoueur);
00152             }
00153         }
00154     }
00155
00156 }

```

## 8.5 Référence du fichier ActiviteFinPartie.java

Déclaration de la classe ActiviteFinPartie.

### Classes

— class [projet.lasalle84.darts.ActiviteFinPartie](#)

### Paquetages

— package [projet.lasalle84.darts](#)

#### 8.5.1 Description détaillée

Déclaration de la classe ActiviteFinPartie.

### Auteur

Menella Erwan

Définition dans le fichier [ActiviteFinPartie.java](#).

## 8.6 ActiviteFinPartie.java

```

00001 package projet.lasalle84.darts;
00002
00003 import androidx.appcompat.app.AppCompatActivity;
00004
00005 import android.os.Bundle;
00006 import android.util.Log;
00007 import android.view.View;
00008 import android.widget.AdapterView;
00009 import android.widget.Button;
00010 import android.widget.ListView;
00011 import android.widget.TextView;
00012
00013 import java.util.ArrayList;
00014 import java.util.Comparator;
00015 import java.util.List;
00016
00022 public class ActiviteFinPartie extends AppCompatActivity implements View.OnClickListener {
00023
00027     private final String TAG = "IHMFinPartie";
00031     private TextView texteGagnant = null;           //<! Widget texte avec le nom du gagnant
00032     private ListView classement = null;           //<! Widget avec la liste des joueurs avec
leur score
00033     private Button partieFini = null;           //<! Bouton pour fermer la page
00034     private ArrayList<Joueur> lesJoueur = null;   //<! Conteneur avec des objets Joueur
00035     private String nomGagnant = null;           //<! Nom du Gagnant
00036     private ArrayAdapter<String> adapteur = null; //<! Adapteur avec le classement des joueurs
00037
00044     @Override
00045     protected void onCreate(Bundle savedInstanceState) {
00046         super.onCreate(savedInstanceState);
00047         Log.d(TAG, "onCreate()");
00048         setContentView(R.layout.activite_fin_partie);
00049         recupererWidgets();
00050         initialiserWidgets();
00051         recupererInfomationPartie();
00052         trierLesJoueurs();
00053         afficherJoueurs();
00054     }
00055
00061     @Override
00062     public void onClick(View element)
00063     {
00064         Log.d(TAG, "onClick() " + element.toString());
00065
00066         if(element == partieFini)
00067         {
00068             super.finish();
00069         }
00070     }
00071
00077     public void recupererWidgets()
00078     {
00079         Log.d(TAG, "recupererWidgets()");
00080         texteGagnant = (TextView) findViewById(R.id.texteGagnant);
00081         classement = (ListView) findViewById(R.id.ResultaPartie);
00082         partieFini = (Button) findViewById(R.id.boutonFin);
00083     }
00084
00085
00091     public void initialiserWidgets()
00092     {
00093         Log.d(TAG, "initialiserWidgets()");
00094         partieFini.setOnClickListener(this);
00095     }
00096
00102     public void recupererInfomationPartie()
00103     {
00104         Log.d(TAG, "recupererInfomationPartie()");
00105         lesJoueur = (ArrayList<Joueur>) getIntent().getSerializableExtra("joueurs");
00106         nomGagnant = getIntent().getStringExtra("gagnant");
00107     }
00108
00109
00115     public void trierLesJoueurs()
00116     {
00117         Log.d(TAG, "trierLesJoueurs()");
00118         lesJoueur.sort(Comparator.comparingInt(Joueur::getScore));
00119     }
00120
00121
00127     public void afficherJoueurs()
00128     {
00129         Log.d(TAG, "afficherJoueurs()");
00130         adapteur = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
00131         for (int i = 0; i < lesJoueur.size(); i++)
00132         {
00133             adapteur.add(lesJoueur.get(i).getNom() + " : " + lesJoueur.get(i).getScore());
00134         }

```

```

00135         classement.setAdapter(adapteur);
00136
00137         texteGagnant.setText(nomGagnant + " a gagné la partie !");
00138     }
00139 }

```

## 8.7 Référence du fichier ActivitePartie.java

### Classes

- class [projet.lasalle84.darts.ActivitePartie](#)  
Déclaration de la classe [ActivitePartie](#).

### Paquetages

- package [projet.lasalle84.darts](#)

## 8.8 ActivitePartie.java

```

00001 package projet.lasalle84.darts;
00007 import androidx.appcompat.app.AppCompatActivity;
00008
00009 import android.bluetooth.BluetoothAdapter;
00010 import android.content.Intent;
00011 import android.os.Bundle;
00012 import android.os.Handler;
00013 import android.os.Message;
00014 import android.os.Parcelable;
00015 import android.util.Log;
00016 import android.view.View;
00017 import android.widget.AdapterView;
00018 import android.widget.Button;
00019 import android.widget.ListView;
00020 import android.widget.TextView;
00021 import android.widget.Toast;
00022
00023 import java.io.Serializable;
00024 import java.util.ArrayList;
00025 import java.util.Iterator;
00026 import java.util.List;
00027
00032 public class ActivitePartie extends AppCompatActivity implements View.OnClickListener
00033 {
00037     private final static String TAG = "IHMPartie";
00038     private final static int REQUEST_CODE_ENABLE_BLUETOOTH = 0;
00039
00042     private Button boutonTirManque = null;
00043     private Button boutonLancerPartie = null;
00044     private ArrayList<Joueur> mesJoueurs = null;
00045     private BluetoothAdapter bluetoothAdapter = null;
00046     private Partie maPartie = null;
00047     private TypeJeu modeJeu = null;
00048     private ArrayAdapter<String> adapteur = null;
00049     private ListView listJoueur = null;
00050     private TextView affichageImpact = null;
00051     private Button boutonPause = null;
00052     private boolean AfficheRegle;
00053
00060     @Override
00061     protected void onCreate(Bundle savedInstanceState)
00062     {
00063         Log.d(TAG, "onCreate()");
00064         super.onCreate(savedInstanceState);
00065         setContentView(R.layout.activity_partie);
00066         recupererWidgets();
00067         initialiserWidgets();
00068         activerBluetooth();
00069         ChargerLesParametre();
00070         initialiserListeJoueur();
00071         maPartie = new Partie(mesJoueurs, modeJeu, AfficheRegle, bluetoothAdapter,
handlerUI);
00072     }
00073
00079     public void ChargerLesParametre()
00080     {

```

```

00081     mesJoueurs = new ArrayList<Joueur>();
00082     mesJoueurs = (ArrayList<Joueur>) getIntent().getSerializableExtra("LesJoueurs");
00083     Iterator<Joueur> it = mesJoueurs.iterator();
00084
00085     while(it.hasNext())
00086     {
00087         Log.d(TAG, "le joueur " + it.next().getNom() + "est chargé");
00088     }
00089     modeJeu = new TypeJeu(getIntent().getIntExtra("TypeMode",0));
00090     AfficheRegle = getIntent().getBooleanExtra("AfficheRegle", true);
00091 }
00092
00099 @Override
00100 public void onClick(View element)
00101 {
00102     Log.d(TAG, "onClick() " + element.toString());
00103
00104     if (element == boutonLancerPartie)
00105     {
00106         new Thread()
00107         {
00108             @Override
00109             public void run() {
00110                 super.run();
00111                 maPartie.demarrer();
00112             }
00113         }.start();
00114         boutonLancerPartie.setVisibility(View.INVISIBLE);
00115         boutonTirManque.setVisibility(View.VISIBLE);
00116     }
00117     else if (element == boutonTirManque)
00118     {
00119         maPartie.cibleManquer();
00120     }
00121     else if (element == boutonPause)
00122     {
00123         if (boutonPause.getText().equals("Pause"))
00124         {
00125             boutonPause.setText("Reprendre");
00126         }
00127         else
00128         {
00129             boutonPause.setText("Pause");
00130         }
00131     }
00132 }
00133
00134 }
00135 public void activerBluetooth()
00136 {
00137     bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
00138     if (bluetoothAdapter == null)
00139     {
00140         Toast.makeText(getApplicationContext(), "Bluetooth non activé !", Toast.LENGTH_SHORT).show();
00141     }
00142     else
00143     {
00144         if (!bluetoothAdapter.isEnabled())
00145         {
00146             Toast.makeText(getApplicationContext(), "Bluetooth non activé !", Toast.LENGTH_SHORT).show(
00147 );
00148             Intent activeBlueTooth = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
00149             startActivityForResult(activeBlueTooth, REQUEST_CODE_ENABLE_BLUETOOTH);
00150         }
00151         else
00152         {
00153             Toast.makeText(getApplicationContext(), "Bluetooth activé", Toast.LENGTH_SHORT).show();
00154         }
00155     }
00156 }
00157
00164 public void recupererWidgets()
00165 {
00166     Log.d(TAG, "recupererWidgets()");
00167     boutonTirManque = (Button) findViewById(R.id.TirManque);
00168     boutonLancerPartie = (Button) findViewById(R.id.LancerPartie);
00169     listJoueur = (ListView) findViewById(R.id.MesJoueur);
00170     affichageImpact = (TextView) findViewById(R.id.Impact);
00171     boutonPause = (Button) findViewById(R.id.Pause);
00172 }
00173
00180 public void initialiserWidgets()
00181 {
00182     Log.d(TAG, "initialiserWidgets()");
00183     boutonTirManque.setOnClickListener(this);
00184     boutonTirManque.setVisibility(View.INVISIBLE);
00185     boutonLancerPartie.setVisibility(View.INVISIBLE);
00186     boutonLancerPartie.setOnClickListener(this);
00187     boutonPause.setVisibility(View.INVISIBLE); // Fonction Pause non-fonctionnel
00188     boutonPause.setOnClickListener(this);

```

```

00189     }
00190
00191     @Override
00192     protected void onActivityResult(int requestCode, int resultCode, Intent data)
00193     {
00194         Log.d(TAG, "onActivityResult ()");
00195         super.onActivityResult(requestCode, resultCode, data);
00196         if (requestCode != REQUEST_CODE_ENABLE_BLUETOOTH)
00197             return;
00198         if (resultCode == RESULT_OK)
00199         {
00200             Toast.makeText(getApplicationContext(), "Bluetooth activé", Toast.LENGTH_SHORT).show();
00201         }
00202         else
00203         {
00204             Toast.makeText(getApplicationContext(), "Bluetooth non activé !", Toast.LENGTH_SHORT).show();
00205         }
00206     }
00207
00208     final private Handler handlerUI = new Handler()
00209     {
00210         @Override
00211         public void handleMessage(Message msg)
00212         {
00213             super.handleMessage(msg);
00214             Bundle b = msg.getData();
00215
00216             switch (b.getInt("action"))
00217             {
00218                 case Partie.JOUEUR_SUIVANT:
00219                     Log.d(TAG, "JOUEUR_SUIVANT " + "Joueur: " + b.getString("joueur"));
00220                     AfficheQuiDoitJouer(b.getString("joueur"));
00221                     break;
00222                 case Partie.SET_SCORE:
00223                     Log.d(TAG, "SET_SCORE " + "Joueur: " + b.getString("joueur") + " Score: " + b.getInt("
00224 score"));
00225                     actualiserScoreIHM(b.getString("joueur"), b.getInt("score"));
00226                     break;
00227                 case Partie.IMPACT:
00228                     Log.d(TAG, "IMPACT " + "Joueur: " + b.getString("joueur") + " TypePoint: " + b.getInt("
00229 typePoint") + " NumeroCible: " + b.getInt("numeroCible"));
00230                     b.getInt("typePoint");
00231                     b.getInt("numeroCible");
00232                     afficherImpact ( b.getInt("typePoint"), b.getInt("numeroCible"));
00233                     break;
00234                 case Partie.GAGNANT:
00235                     Log.d(TAG, "GAGNANT " + b.getString("gagnant"));
00236                     afficheGagnant (b.getString("gagnant"));
00237                     break;
00238                 case Partie.CONNEXION_CIBLE:
00239                     Log.d(TAG, "CONNEXION_CIBLE ");
00240                     boutonLancerPartie.setVisibility(View.VISIBLE);
00241                     break;
00242             }
00243         }
00244     };
00245
00246     public void initialiserListeJoueur ()
00247     {
00248         Log.d(TAG, "initialiserListeJoueur ()");
00249         adapteur = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item);
00250         for (int i = 0; i < mesJoueurs.size(); i++)
00251         {
00252             adapteur.add(mesJoueurs.get(i).getNom());
00253         }
00254         listeJoueur.setAdapter(adapteur);
00255     }
00256
00257     public void actualiserScoreIHM(String joueur, int score)
00258     {
00259         Log.d(TAG, "actualiserScoreIHM ()");
00260         int idJoueur = 0;
00261
00262         for (int i = 0; i < mesJoueurs.size(); i++)
00263         {
00264             if (mesJoueurs.get(i).getNom().equals(joueur))
00265             {
00266                 idJoueur = i;
00267             }
00268         }
00269         mesJoueurs.get(idJoueur).setScore(score);
00270         String chaineCaractere = mesJoueurs.get(idJoueur).getNom() + ": " + mesJoueurs.get(idJoueur).
00271 getScore ();
00272         adapteur.remove(adapteur.getItem(idJoueur));
00273         adapteur.insert(chaineCaractere, idJoueur);
00274     }
00275
00276     public void AfficheQuiDoitJouer (String joueur)
00277     {
00278         Log.d(TAG, "AfficheQuiDoitJouer ()");
00279         String chaineCaractere;

```

```

00301     for (int i = 0; i < mesJoueurs.size(); i++)
00302     {
00303         if (mesJoueurs.get(i).getNom().equals(joueur))
00304         {
00305             chaineCaractere = "-> " + mesJoueurs.get(i).getNom() + ": " + mesJoueurs.get(i).getScore();
00306         }
00307         else {
00308             chaineCaractere = mesJoueurs.get(i).getNom() + ": " + mesJoueurs.get(i).getScore();
00309         }
00310         adaptateur.remove(adaptateur.getItem(i));
00311         adaptateur.insert(chaineCaractere, i);
00312     }
00313 }
00314 afficheImpact.setText("");
00315 }
00316
00323 public void afficherImpact(int typeImpact, int numeroImpact)
00324 {
00325     Log.d(TAG, "afficherImpact()");
00326     String lettreTypeImpact = new String();
00327     switch (typeImpact)
00328     {
00329         case 0:
00330             lettreTypeImpact = "MISS";
00331             break;
00332         case 1:
00333             lettreTypeImpact = "S";
00334             break;
00335         case 2:
00336             lettreTypeImpact = "D";
00337             break;
00338         case 3:
00339             lettreTypeImpact = "T";
00340             break;
00341     }
00342     if (lettreTypeImpact == "MISS")
00343     {
00344         afficheImpact.setText(afficheImpact.getText().toString().concat("MISS "));
00345     }
00346     else
00347     {
00348         afficheImpact.setText(afficheImpact.getText().toString().concat(lettreTypeImpact + Integer.
toString(numeroImpact) + " "));
00349     }
00350 }
00351
00358 public void afficheGagnant(String gagnant)
00359 {
00360     Log.d(TAG, "afficheGagnant()");
00361     Intent IHMGagnant = new Intent(ActivitePartie.this,
ActiviteFinPartie.class);
00362     List<Joueur>lesJoueur = new ArrayList<Joueur>();
00363     for (int i = 0; i < mesJoueurs.size(); i++)
00364     {
00365         if (mesJoueurs.get(i).getNom().equals(gagnant))
00366         {
00367             IHMGagnant.putExtra("gagnant", mesJoueurs.get(i).getNom());
00368         }
00369         else
00370         {
00371             lesJoueur.add(mesJoueurs.get(i));
00372         }
00373     }
00374     IHMGagnant.putExtra("joueurs", (Serializable) lesJoueur);
00375     startActivity(IHMGagnant);
00376     finish();
00377 }
00378
00379
00380 @Override
00381 public void finish()
00382 {
00383     super.finish();
00384     maPartie.deconnecterPeripheriquesBluetooth();
00385 }
00386 }
00387

```

## 8.9 Référence du fichier Changelog.md

### 8.10 Changelog.md

```

00001 \page page_changelog Changelog
00002
00003 r1 | www-data | 2020-02-01 15:03:29 +0100 (sam. 01 févr. 2020) | 1 ligne
00004
00005 Creating initial repository structure

```

## 8.11 Référence du fichier ExampleInstrumentedTest.java

### Classes

— class [projet.lasalle84.darts.ExampleInstrumentedTest](#)

### Paquetages

— package [projet.lasalle84.darts](#)

## 8.12 ExampleInstrumentedTest.java

```
00001 package projet.lasalle84.darts;
00002
00003 import android.content.Context;
00004
00005 import androidx.test.platform.app.InstrumentationRegistry;
00006 import androidx.test.ext.junit.runners.AndroidJUnit4;
00007
00008 import org.junit.Test;
00009 import org.junit.runner.RunWith;
00010
00011 import static org.junit.Assert.*;
00012
00018 @RunWith(AndroidJUnit4.class)
00019 public class ExampleInstrumentedTest {
00020     @Test
00021     public void useAppContext() {
00022         // Context of the app under test.
00023         Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();
00024
00025         assertEquals("projet.lasalle84.darts", appContext.getPackageName());
00026     }
00027 }
```

## 8.13 Référence du fichier ExampleUnitTest.java

### Classes

— class [projet.lasalle84.darts.ExampleUnitTest](#)

### Paquetages

— package [projet.lasalle84.darts](#)

## 8.14 ExampleUnitTest.java

```
00001 package projet.lasalle84.darts;
00002
00003 import org.junit.Test;
00004
00005 import static org.junit.Assert.*;
00006
00012 public class ExampleUnitTest {
00013     @Test
00014     public void addition_isCorrect() {
00015         assertEquals(4, 2 + 2);
00016     }
00017 }
```

## 8.15 Référence du fichier Joueur.java

Déclaration de la classe Joueur.

## Classes

- class [projet.lasalle84.darts.Joueur](#)  
Déclaration de la classe *Joueur*.

## Paquetages

- package [projet.lasalle84.darts](#)

### 8.15.1 Description détaillée

Déclaration de la classe *Joueur*.

#### Auteur

Menella Erwan

Définition dans le fichier [Joueur.java](#).

### 8.16 Joueur.java

```

00001 package projet.lasalle84.darts;
00002
00010 import android.util.Log;
00011 import java.io.Serializable;
00012
00013
00018 public class Joueur implements Serializable
00019 {
00020     private static final String TAG = "Joueur" ;
00021     private String nom;
00022     private int score;
00023     private boolean estEliminer;
00024
00031     public Joueur(String nom)
00032     {
00033         this.nom = nom;
00034         this.score = 0;
00035         estEliminer = false;
00036     }
00037
00044     public String getNom()
00045     {
00046         return nom;
00047     }
00048
00055     public void setNom(String nom)
00056     {
00057         this.nom = nom;
00058     }
00059
00066     public int getScore()
00067     {
00068         return score;
00069     }
00070
00077     public void setScore(int score) {
00078         Log.d(TAG, this.nom + " setScore " + score);
00079         this.score = score;
00080     }
00081
00088     public boolean retirerPoint(int scoreLancer, Partie maPartie)
00089     {
00090         if (this.score - scoreLancer >= 0)
00091         {
00092             this.score = this.score - scoreLancer;
00093             if (this.score == 1 && maPartie.getTypeJeu().estDoubleOut())
00094             {
00095                 Log.d(TAG, this.getNom() + " est éliminé");
00096                 estEliminer = true;
00097             }
00098             return true;
00099         }
00100         return false;

```

```
00101     }
00102
00109     public void setEstEliminer(boolean estEliminer)
00110     {
00111         this.estEliminer = estEliminer;
00112     }
00113
00119     public boolean estEliminer()
00120     {
00121         return estEliminer;
00122     }
00123
00124 }
```

## 8.17 Référence du fichier MainActivity.java

Déclaration de la classe MainActivity.

### Classes

- class [projet.lasalle84.darts.MainActivity](#)  
*Déclaration de la classe MainActivity.*

### Paquetages

- package [projet.lasalle84.darts](#)

#### 8.17.1 Description détaillée

Déclaration de la classe MainActivity.

### Auteur

Menella Erwan

Définition dans le fichier [MainActivity.java](#).

## 8.18 MainActivity.java

```
00001 package projet.lasalle84.darts;
00002
00003 import androidx.appcompat.app.AppCompatActivity;
00004
00005 import android.content.Intent;
00006 import android.os.Bundle;
00007 import android.util.Log;
00008 import android.view.View;
00009 import android.widget.Button;
00010
00021 public class MainActivity extends AppCompatActivity implements View.OnClickListener
00022 {
00026     private static final String TAG = "IHMEcranPrincipal";
00027
00031     private Button boutonDemarrerPartie;
00032     private Button boutonGererJoueurs;
00033     private Button boutonVoirHistorique;
00034
00035
00042     @Override
00043     protected void onCreate(Bundle savedInstanceState)
00044     {
00045         super.onCreate(savedInstanceState);
00046         Log.d(TAG, "onCreate()");
00047
00048         // Associer le layout à l'activité
00049         setContentView(R.layout.activity_main);
```

```

00050
00051     recupererWidgets();
00052
00053     initialiserWidgets();
00054 }
00055
00062 @Override
00063 public void onClick(View element)
00064 {
00065     Log.d(TAG, "onClick()");
00066
00067     if(element == boutonDemarrerPartie)
00068     {
00069         Log.d(TAG, "clic boutonDemarrerPartie");
00070         Intent activiteCreerPartie = new Intent(MainActivity.this,
ActiviteCreerPartie.class);
00071         startActivity(activiteCreerPartie);
00072     }
00073
00074     else if(element == boutonGererJoueurs)
00075     {
00076         Log.d(TAG, "clic boutonGererJoueurs");
00077     }
00078     else if(element == boutonVoirHistorique)
00079     {
00080         Log.d(TAG, "clic boutonVoirHistorique");
00081     }
00082 }
00083
00090 private void recupererWidgets()
00091 {
00092     Log.d(TAG, "recupererWidgets()");
00093
00094     boutonDemarrerPartie = (Button)findViewById(R.id.boutonDemarrerPartie);
00095     boutonGererJoueurs = (Button)findViewById(R.id.boutonGererJoueurs);
00096     boutonVoirHistorique = (Button)findViewById(R.id.boutonVoirHistorique);
00097 }
00098
00105 private void initialiserWidgets()
00106 {
00107     Log.d(TAG, "initialiserWidgets()");
00108
00109     // Rend les boutons cliquables
00110     boutonDemarrerPartie.setOnClickListener(this);
00111     boutonGererJoueurs.setOnClickListener(this);
00112     boutonVoirHistorique.setOnClickListener(this);
00113
00114     boutonGererJoueurs.setVisibility(View.INVISIBLE);
00115     boutonVoirHistorique.setVisibility(View.INVISIBLE);
00116 }
00117 }
00118

```

## 8.19 Référence du fichier Partie.java

Déclaration de la classe Partie.

### Classes

- class [projet.lasalle84.darts.Partie](#)  
Déclaration de la classe [Partie](#).

### Paquetages

- package [projet.lasalle84.darts](#)

#### 8.19.1 Description détaillée

Déclaration de la classe Partie.

### Auteur

Menella Erwan

Définition dans le fichier [Partie.java](#).

## 8.20 Partie.java

```

00001 package projet.lasalle84.darts;
00002
00003 import android.bluetooth.BluetoothAdapter;
00004 import android.bluetooth.BluetoothDevice;
00005 import android.os.Bundle;
00006 import android.os.Handler;
00007 import android.os.Message;
00008 import android.util.Log;
00009
00010 import java.util.ArrayList;
00011 import java.util.Iterator;
00012 import java.util.Set;
00013
00014 import static android.os.SystemClock.sleep;
00015
00026 public class Partie
00027 {
00031     private final static String TAG = "Partie";
00032     private final static String NOM_PERIPHERIQUE_BLUETOOTH_ECRAN = "
ecran-darts";
00033     private final static String NOM_PERIPHERIQUE_BLUETOOTH_CIBLE = "
impact-darts";
00034     public final static int NB_FLECHETTE = 3;
00038     public final static int JOUEUR_SUIVANT = 0;
00039     public final static int SET_SCORE = 1;
00040     public final static int IMPACT = 2;
00041     public final static int GAGNANT = 3;
00042     public final static int CONNEXION_CIBLE = 4;
00046     private int nbJoueurs;
00047     private int nbManche;
00048     private TypeJeu typeJeu;
00049     private ArrayList<Joueur> lesJoueurs;
00050     private BluetoothAdapter bluetoothAdapter;
00051     private Set<BluetoothDevice> devices;
00052     private Peripherique ecran = null;
00053     private Peripherique cible = null;
00054     private Handler handlerUI = null;
00055     private int impact[] = null;
00056     private boolean impactEstRecuperer;
00057     private boolean estFini = false;
00058     private boolean estDoubleImpact = false;
00059     private boolean afficheRegle = true;
00060
00068     public Partie(ArrayList<Joueur> lesJoueurs, TypeJeu typeJeu, boolean AfficheRegle,
BluetoothAdapter bluetoothAdapter, Handler handlerUI)
00069     {
00070         this.lesJoueurs = lesJoueurs;
00071         this.typeJeu = typeJeu;
00072         this.bluetoothAdapter = bluetoothAdapter;
00073         this.handlerUI = handlerUI;
00074         this.nbManche = 0;
00075         this.nbJoueurs = this.lesJoueurs.size();
00076         impact = new int[2];
00077         this.afficheRegle = AfficheRegle;
00078         recupererPeripheriques();
00079         connecterPeripheriquesBluetooth();
00080     }
00081
00082
00089     public void recupererPeripheriques ()
00090     {
00091         Log.d(TAG, "recupererPeripheriques()");
00092         devices = bluetoothAdapter.getBondedDevices();
00093     }
00094
00100     public void connecterPeripheriquesBluetooth ()
00101     {
00102         Log.d(TAG, "connecterPeripheriquesBluetooth()");
00103
00104         String nomPeripherique = NOM_PERIPHERIQUE_BLUETOOTH_ECRAN;
00105         for (BluetoothDevice device : devices)
00106         {
00107             if(device.getName().contains(nomPeripherique))
00108             {
00109                 ecran = new Peripherique(device, handlerBluetooth);
00110                 Log.d(TAG, "Adresse du périphérique raspberrypi " + device.getAddress());
00111                 break;
00112             }
00113         }
00114
00115         nomPeripherique = NOM_PERIPHERIQUE_BLUETOOTH_CIBLE;
00116         for (BluetoothDevice device : devices)
00117         {
00118             if(device.getName().contains(nomPeripherique))
00119             {
00120                 cible = new Peripherique(device, handlerBluetooth);
00121                 Log.d(TAG, "Adresse du périphérique darts " + device.getAddress());
00122                 break;

```

```

00123     }
00124     }
00125
00126     if(ecran != null)
00127         ecran.connecter();
00128     if(cible != null)
00129         cible.connecter();
00130 }
00131
00138 public void deconnecterPeripheriquesBluetooth()
00139 {
00140     Log.d(TAG, "deconnecterPeripheriquesBluetooth()");
00141     if (ecran != null)
00142     {
00143         ecran.deconnecter();
00144     }
00145     if (cible != null)
00146     {
00147         cible.deconnecter();
00148     }
00149 }
00150 }
00151
00158 public void envoyerTrame(Peripherique peripherique, String trame)
00159 {
00160     if(peripherique != null)
00161     {
00162         Log.d(TAG,"envoyerTrame() " + peripherique.getNom());
00163         peripherique.envoyer(trame);
00164     }
00165 }
00172 public void demarrer() {
00173     Log.d(TAG, "demarrer()");
00174     initialiserPartie();
00175
00176     do
00177     {
00178         nbManche++;
00179         Iterator<Joueur> it = lesJoueurs.iterator();
00180         Log.d(TAG, "Manche numéro " + nbManche);
00181         while (it.hasNext())
00182         {
00183             Joueur monJoueur = it.next();
00184             Log.d(TAG, "c'est le tour à " + monJoueur.getNom());
00185             actualiserJoueurIHM(monJoueur.getNom());
00186             int pointVolley = 0;
00187
00188             for(int i = 0; i < NB_FLECHETTE; i++)
00189             {
00190                 if (monJoueur.estEliminer())
00191                 {
00192                     i = NB_FLECHETTE;
00193                 }
00194                 else
00195                 {
00196                     impactEstRecuperer = false;
00197                     attendreImpact();
00198                     pointVolley += impact[0]*impact[1];
00199                     if(!monJoueur.retirerPoint(pointVolley, this))
00200                     {
00201                         i = NB_FLECHETTE;
00202                         pointVolley = 0;
00203                     }
00204                     else if (monJoueur.getScore() == 0 && !typeJeu.
estDoubleOut())
00205                     {
00206                         envoyerGagnantIHM(monJoueur);
00207                         i = NB_FLECHETTE;
00208                         estFini = true;
00209                     }
00210                     else if (monJoueur.getScore() == 0 && typeJeu.
estDoubleOut() && estDoubleImpact)
00211                     {
00212                         envoyerGagnantIHM(monJoueur);
00213                         i = NB_FLECHETTE;
00214                         estFini = true;
00215                     }
00216                 }
00217             }
00218         }
00219         actualiserScoreIHM(monJoueur, monJoueur.
getScore());
00220     }
00221     }while (!estFini);
00222 }
00223
00230 public void initialiserPartie() {
00231     Log.d(TAG, "initialiserPartie()");
00232     Iterator<Joueur> it = lesJoueurs.iterator();
00233     String nomJoueurTrame = new String();
00234     while(it.hasNext())

```

```

00235     {
00236         Joueur monJoueur = it.next();
00237         monJoueur.setScore(typeJeu.getPointDepart());
00238         actualiserScoreIHM(monJoueur, typeJeu.
getPointDepart());
00239         nomJoueurTrame = nomJoueurTrame.concat(monJoueur.getNom() + ";");
00240     }
00241     int afficheRegleInt = afficheRegle ? 1 : 0;
00242     envoyerTrame(ecran, "$DARTS;START;" + typeJeu.getTypeJeu() + ";" +
afficheRegleInt + ";" + lesJoueurs.size() + ";" + nomJoueurTrame + "\r\n");
00243 }
00244
00245
00246     private Handler handlerBluetooth = new Handler()
00247     {
00248         @Override
00249         public void handleMessage(Message msg)
00250         {
00251             super.handleMessage(msg);
00252             Bundle b = msg.getData();
00253
00254             switch(b.getInt("etat"))
00255             {
00256                 case Peripherique.CODE_ERREUR_CONNECTER:
00257                     Log.d(TAG, "<Bluetooth> Erreur " + b.getString("nom") + " [" + b.getString("adresse") +
"] connecter");
00258                     connecterPeripheriquesBluetooth();
00259                     break;
00260                 case Peripherique.CODE_ERREUR_RECEVOIR:
00261                     Log.d(TAG, "<Bluetooth> Erreur " + b.getString("nom") + " [" + b.getString("adresse") +
"] envoyer");
00262                     connecterPeripheriquesBluetooth();
00263                     break;
00264                 case Peripherique.CODE_ERREUR_ENVOYER:
00265                     Log.d(TAG, "<Bluetooth> Erreur " + b.getString("nom") + " [" + b.getString("adresse") +
"] envoyer");
00266                     connecterPeripheriquesBluetooth();
00267                     break;
00268                 case Peripherique.CODE_CONNEXION:
00269                     Log.d(TAG, "<Bluetooth> Connexion " + b.getString("nom") + " [" + b.getString("adresse")
+ "] ok");
00270                     envoyerConnexionIHM(b.getString("nom"));
00271                     break;
00272                 case Peripherique.CODE_RECEPTION:
00273                     String donnees = b.getString("donnees");
00274                     if(donnees.contains("\r\n"))
00275                     {
00276                         Log.d(TAG, "<Bluetooth> Données reçues " + b.getString("nom") + " [" + b.getString("
adresse") + "] : " + donnees.replace("\r\n", " "));
00277                         traitementTrame(donnees);
00278                     }
00279                     else
00280                     {
00281                         Log.d(TAG, "<Bluetooth> Données reçues " + b.getString("nom") + " [" + b.getString("
adresse") + "] : " + donnees);
00282                         traitementTrame(donnees);
00283                     }
00284                     break;
00285                 case Peripherique.CODE_DECONNEXION:
00286                     Log.d(TAG, "<Bluetooth> Déconnexion " + b.getString("nom") + " [" + b.getString("adresse
") + "] ok");
00287                     break;
00288                 default:
00289                     Log.d(TAG, "<Bluetooth> code état inconnu ! ");
00290             }
00291         }
00292     };
00293
00300     public void attendreImpact() {
00301         Log.d(TAG, "attendreImpact()");
00302         while (!impactEstRecuperer)
00303         {
00304             sleep(1000);
00305         }
00306         envoyerTrame(ecran, "$DART;GAME;" + impact[0] + ";" + impact[1] + "\r\n");
00307         impactIHM(impact[0], impact[1]);
00308
00309         if (impact[0] == 2)
00310         {
00311             estDoubleImpact = true;
00312         }
00313         else
00314         {
00315             estDoubleImpact = false;
00316         }
00317
00318         Log.d(TAG, "Type cible: " + impact[0] + "Numero Cible:" + impact[1]);
00319
00320     }
00321 }
00322

```

```

00329 public void traitementTrame(String trame)
00330 {
00331     trame = trame.replace("\r\n", "");
00332     String[] trameDecoupe = trame.split(";",4);
00333     Bundle bundle = new Bundle();
00334
00335     switch (trameDecoupe[1])
00336     {
00337         case "GAME":
00338             Log.d(TAG, "traitementTrame: GAME");
00339             impact[0] = Integer.parseInt(trameDecoupe[2]);
00340             impact[1] = Integer.parseInt(trameDecoupe[3]);
00341             impactEstRecuperer = true;
00342
00343             break;
00344         case "PAUSE":
00345             //TODO PAUSE
00346             break;
00347         case "PLAY":
00348             //TODO PLAY
00349             break;
00350         case "STOP":
00351             //TODO STOP
00352             break;
00353         case "RESET":
00354             //TODO RESET
00355             break;
00356         case "HEARTBEAT ":
00357             //TODO HEARTBEAT
00358             break;
00359         case "ACK":
00360             //TODO ACK
00361             break;
00362     }
00363 }
00364
00371 public void actualiserScoreIHM(Joueur monJoueur, int score)
00372 {
00373     Log.d(TAG, "actualiserScoreIHM()");
00374     Message msg = Message.obtain();
00375     Bundle b = new Bundle();
00376     b.putInt("action", SET_SCORE);
00377     b.putString("joueur", monJoueur.getNom());
00378     b.putInt("score", score);
00379     msg.setData(b);
00380     handlerUI.sendMessage(msg);
00381 }
00382
00389 public void actualiserJoueurIHM(String monJoueur)
00390 {
00391     Log.d(TAG, "actualiserJoueurIHM()");
00392     Message msg = Message.obtain();
00393     Bundle b = new Bundle();
00394     b.putInt("action", JOUEUR_SUIVANT);
00395     b.putString("joueur", monJoueur);
00396     msg.setData(b);
00397     handlerUI.sendMessage(msg);
00398 }
00399
00406 public void impactIHM(int typePoint, int numeroCible)
00407 {
00408     Log.d(TAG, "impactIHM()");
00409     Message msg = Message.obtain();
00410     Bundle b = new Bundle();
00411     b.putInt("action", IMPACT);
00412     b.putInt("typePoint", typePoint);
00413     b.putInt("numeroCible", numeroCible);
00414     msg.setData(b);
00415     handlerUI.sendMessage(msg);
00416 }
00417
00424 public void envoyerGagnantIHM(Joueur monJoueur) {
00425     Log.d(TAG, "envoyerGagnantIHM()");
00426     estFini = true;
00427     deconnecterPeripheriquesBluetooth();
00428     Message msg = Message.obtain();
00429     Bundle b = new Bundle();
00430     b.putInt("action", GAGNANT);
00431     b.putString("gagnant", monJoueur.getNom());
00432     msg.setData(b);
00433     handlerUI.sendMessage(msg);
00434 }
00435
00436 public void cibleManquer()
00437 {
00438     impact[0] = 0;
00439     impact[1] = 0;
00440     impactEstRecuperer = true;
00441 }
00442
00443 public void envoyerConnexionIHM(String peripherique)

```

```
00444     {
00445         if (peripherique.equals(NOM_PERIPHERIQUE_BLUETOOTH_CIBLE))
00446         {
00447             Log.d(TAG, "envoyerConnexionIHM()");
00448             Message msg = Message.obtain();
00449             Bundle b = new Bundle();
00450             b.putInt("action", CONNEXION_CIBLE);
00451             msg.setData(b);
00452             handlerUI.sendMessage(msg);
00453         }
00454     }
00455
00456     public void pause()
00457     {
00458         //TODO pause()
00459     }
00460
00461     public void reprendre()
00462     {
00463         //TODO reprendre()
00464     }
00465
00466     public TypeJeu getTypeJeu() {
00467         return typeJeu;
00468     }
00469 }
```

## 8.21 Référence du fichier Peripherique.java

Déclaration de la classe Peripherique.

### Classes

- class [projet.lasalle84.darts.Peripherique](#)  
*Déclaration de la classe [Peripherique](#).*

### Paquetages

- package [projet.lasalle84.darts](#)

#### 8.21.1 Description détaillée

Déclaration de la classe Peripherique.

### Auteur

Menella Erwan

Définition dans le fichier [Peripherique.java](#).

## 8.22 Peripherique.java

```

00001 package projet.lasalle84.darts;
00002
00003 import android.bluetooth.BluetoothDevice;
00004 import android.bluetooth.BluetoothSocket;
00005 import android.os.Bundle;
00006 import android.os.Handler;
00007 import android.os.Message;
00008 import android.util.Log;
00009
00010 import java.io.IOException;
00011 import java.io.InputStream;
00012 import java.io.OutputStream;
00013 import java.util.UUID;
00014
00026 public class Peripherique extends Thread
00027 {
00031     private final static String TAG = "Peripherique";
00032     public final static int CODE_CONNEXION = 0;
00033     public final static int CODE_RECEPTION = 1;
00034     public final static int CODE_DECONNEXION = 2;
00035     public final static int CODE_ERREUR_ENVOYER = -1;
00036     public final static int CODE_ERREUR_RECEVOIR = -2;
00037     public final static int CODE_ERREUR_CONNECTER = -3;
00038
00041     private BluetoothDevice device;
00042     private String nom;
00043     private String adresse;
00044     private Handler handler;
00045     private BluetoothSocket socket = null;
00046     private InputStream receiveStream = null;
00047     private OutputStream sendStream = null;
00048     private TReception tReception;
00049
00055     public Peripherique(BluetoothDevice device, Handler handler)
00056     {
00057         Log.d(TAG,"Peripherique() " + device.getName() + "[" + device.getAddress() + "]");
00058         if(device != null)
00059         {
00060             this.device = device;
00061             this.nom = device.getName();
00062             this.adresse = device.getAddress();
00063             this.handler = handler;
00064         }
00065         else
00066         {
00067             this.device = device;
00068             this.nom = "Aucun";
00069             this.adresse = "";
00070             this.handler = handler;
00071         }
00072
00073         try
00074         {
00075             assert device != null;
00076             socket = device.createRfcommSocketToServiceRecord(UUID.fromString("
00077 00001101-0000-1000-8000-00805F9B34FB"));
00078             receiveStream = socket.getInputStream();
00079             sendStream = socket.getOutputStream();
00080         }
00081         catch (IOException e)
00082         {
00083             e.printStackTrace();
00084             socket = null;
00085         }
00086
00087         if(socket != null)
00088             tReception = new TReception(this, handler, receiveStream);
00089
00095     public String getNom()
00096     {
00097         return nom;
00098     }
00099
00105     public String getAdresse()
00106     {
00107         return adresse;
00108     }
00109
00114     public void connecter()
00115     {
00116         Log.d(TAG,"connecter() " + device.getName() + "[" + device.getAddress() + "]");
00117         new Thread()
00118         {
00119             @Override public void run()
00120             {
00121                 try
00122                 {

```

```

00123         socket.connect();
00124
00125         Message msg = Message.obtain();
00126         Bundle b = new Bundle();
00127         b.putString("nom", getNom());
00128         b.putString("adresse", getAdresse());
00129         b.putInt("etat", CODE_CONNEXION);
00130         msg.setData(b);
00131         handler.sendMessage(msg);
00132
00133         // on démarre le thread de réception
00134         tReception.start();
00135         Log.d(TAG,"connexion reussie " + getNom());
00136
00137     }
00138     catch (IOException e)
00139     {
00140         e.printStackTrace();
00141         Log.d(TAG,"erreur connexion " + getNom());
00142         Message msg = Message.obtain();
00143         Bundle b = new Bundle();
00144         b.putString("nom", getNom());
00145         b.putString("adresse", getAdresse());
00146         b.putInt("etat", CODE_ERREUR_CONNECTER);
00147         msg.setData(b);
00148         handler.sendMessage(msg);
00149     }
00150 }
00151 }.start();
00152 }
00153
00158 public boolean deconnecter()
00159 {
00160     Log.d(TAG,"deconnecter() " + device.getName() + "[" + device.getAddress() + "]");
00161     try
00162     {
00163         tReception.arreter();
00164
00165         socket.close();
00166         return true;
00167     }
00168     catch (IOException e)
00169     {
00170         System.out.println("<Socket> error close");
00171         e.printStackTrace();
00172         return false;
00173     }
00174 }
00175
00180 public void envoyer(final String data)
00181 {
00182     Log.d(TAG,"envoyer() " + device.getName() + "[" + device.getAddress() + "]");
00183     if(socket == null)
00184     {
00185         Log.d(TAG,"pas d'envoi");
00186         return;
00187     }
00188     new Thread()
00189     {
00190         @Override public void run()
00191         {
00192             try
00193             {
00194                 if(socket.isConnected())
00195                 {
00196                     sendStream.write(data.getBytes());
00197                     sendStream.flush();
00198                     Log.d(TAG, "envoyer() trame envoyée : " + data);
00199                 }
00200             }
00201             catch (IOException e)
00202             {
00203                 e.printStackTrace();
00204                 Log.d(TAG, "envoyer() Erreur socket write : " + getNom());
00205                 Message msg = Message.obtain();
00206                 Bundle b = new Bundle();
00207                 b.putString("nom", getNom());
00208                 b.putString("adresse", getAdresse());
00209                 b.putInt("etat", CODE_ERREUR_ENVOYER);
00210                 msg.setData(b);
00211                 handler.sendMessage(msg);
00212             }
00213         }
00214     }.start();
00215 }
00216 }

```

## 8.23 Référence du fichier README.md

## 8.24 README.md

```

00001 \mainpage Le projet
00002
00003 \tableofcontents
00004
00005 \section section_tdm Table des matières
00006 - \ref page_README
00007 - \ref page_changelog
00008 - \ref page_about
00009 - \ref page_licence
00010
00011 \section section_infos Informations
00012
00013 \author Erwan Menella <menella.erwan@gmail.com>
00014 \date 2020
00015 \version 0.2
00016 \see https://svn.riouxsvn.com/darts-2020/
00017
00018
00019 \page page_README README
00020
00021 [TOC]
00022
00023 # Projet {#projet}
00024
00025 ## Présentation {#presentation}
00026
00027 Le système DARTS est un système numérique permettant de jouer au jeu de fléchettes électroniques.
00028
00029 Le système DARTS est décomposé en trois modules, dont deux modules sont réalisés par des étudiants IR
00030 :
00031 * Module de gestion de partie (Mobile-DARTS) : les joueurs paramètrent et lancent la partie à
00032 partir d'une application sur un terminal mobile (sous Android) ;
00033
00034 * Module de détection des impacts (Cible-DARTS) : la cible est équipée de capteurs permettant
00035 d'identifier la zone impactée par les fléchettes envoyées par les joueurs ;
00036
00037 * Module de visualisation de partie (Écran-DARTS) : les joueurs, les arbitres et le public
00038 peuvent visualiser en "temps réel" le déroulement de la partie (nombre de manche, point restant dans la manche,
00039 moyenne des volées, ...) sur un écran de télévision.
00040
00041 ## Module de gestion de partie (Mobile-DARTS) {#mobile}
00042
00043 Sur le terminal mobile Android, l'application doit permettre de paramétrer et démarrer une partie.
00044
00045 Pour cela, les joueurs pourront :
00046
00047 * saisir leur nom
00048 * paramétrer la partie :
00049   * le type de jeu : 501 double out, 301 double out, ...
00050   * le nombre de joueurs,
00051   * le nombre de manches gagnantes,
00052 * lancer la partie
00053 * gérer et visualiser le déroulement de la partie :
00054   * changer automatiquement de joueur
00055   * affichage du nombre de points
00056   * renseigner le module Cible d'une fléchette hors cible
00057
00058 Pour démarrer une partie, il est nécessaire d'être connecté en Bluetooth avec le module Cible-DARTS.
00059 La connexion avec le module Écran-DARTS n'est pas indispensable pour le fonctionnement du système DARTS. On
00060 pourra visualiser sur l'interface l'état de connexion de chaque module.
00061
00062 Le terminal mobile possède une base de données permettant de stocker les statistiques des joueurs pour
00063 chaque partie.
00064
00065 ## Informations {#informations}
00066
00067 \author Erwan Menella <menella.erwan@gmail.com>
00068 \date 2020
00069 \version 0.2
00070 \see https://svn.riouxsvn.com/darts-2020/
00071
00072
00073 \page page_about A propos
00074
00075 \author Erwan Menella <menella.erwan@gmail.com>
00076 \date 2020
00077 \version 0.2
00078 \see https://svn.riouxsvn.com/darts-2020/
00079
00080
00081 \page page_licence Licence GPL
00082
00083 This program is free software; you can redistribute it and/or modify
00084 it under the terms of the GNU General Public License as published by
00085 the Free Software Foundation; either version 2 of the License, or
00086 (at your option) any later version.
00087
00088

```

```

00081 This program is distributed in the hope that it will be useful,
00082 but WITHOUT ANY WARRANTY; without even the implied warranty of
00083 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00084 GNU General Public License for more details.
00085
00086 You should have received a copy of the GNU General Public License
00087 along with this program; if not, write to the Free Software
00088 Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

```

## 8.25 Référence du fichier TReception.java

Déclaration de la classe Partie.

### Paquetages

— package [projet.lasalle84.darts](#)

### 8.25.1 Description détaillée

Déclaration de la classe Partie.

### Auteur

Menella Erwan

Définition dans le fichier [TReception.java](#).

## 8.26 TReception.java

```

00001 package projet.lasalle84.darts;
00002
00003 import android.os.Bundle;
00004 import android.os.Handler;
00005 import android.os.Message;
00006 import android.util.Log;
00007
00008 import java.io.BufferedReader;
00009 import java.io.IOException;
00010 import java.io.InputStream;
00011 import java.io.InputStreamReader;
00022 class TReception extends Thread
00023 {
00027     private final String TAG = "TReception";
00028
00031     private Peripherique peripherique;           // Peripherique à écouter
00032     private Handler handler;
00033     private boolean estFini;
00034     private InputStream receiveStream;
00035
00043     TReception(Peripherique peripherique, Handler handler, InputStream flux)
00044     {
00045         Log.d(TAG,"TReception() " + peripherique.getNom() + "[" + peripherique.getAdresse() + "]");
00046         this.peripherique = peripherique;
00047         this.handler = handler;
00048         estFini = false;
00049         receiveStream = flux;
00050     }
00051
00056     @Override
00057     public void run()
00058     {
00059         Log.d(TAG,"début run() " + peripherique.getNom() + "[" + peripherique.getAdresse() + "]");
00060         BufferedReader reception = new BufferedReader(new InputStreamReader(receiveStream));
00061         while(!estFini)
00062         {
00063             try
00064             {
00065                 String trame = "";
00066                 if(reception.ready())
00067             {

```

```

00068         trame = reception.readLine();
00069     }
00070     if(trame.length() > 0)
00071     {
00072         Log.d(TAG, "run() trame reçue : " + trame);
00073         Message msg = Message.obtain();
00074         Bundle b = new Bundle();
00075         b.putString("nom", peripherique.getNom());
00076         b.putString("adresse", peripherique.getAdresse());
00077         b.putInt("etat", Peripherique.CODE_RECEPTION);
00078         b.putString("donnees", trame);
00079         msg.setData(b);
00080         handler.sendMessage(msg);
00081     }
00082 }
00083 catch (IOException e)
00084 {
00085     e.printStackTrace();
00086     Log.d(TAG, "run() Erreur socket read : " + peripherique.getNom());
00087     Message msg = Message.obtain();
00088     Bundle b = new Bundle();
00089     b.putString("nom", peripherique.getNom());
00090     b.putString("adresse", peripherique.getAdresse());
00091     b.putInt("etat", Peripherique.CODE_ERREUR_RECEVOIR);
00092     msg.setData(b);
00093     handler.sendMessage(msg);
00094 }
00095 try
00096 {
00097     Thread.sleep(250);
00098 }
00099 catch (InterruptedException e)
00100 {
00101     e.printStackTrace();
00102 }
00103 }
00104 Message msg = Message.obtain();
00105 Bundle b = new Bundle();
00106 b.putString("nom", peripherique.getNom());
00107 b.putString("adresse", peripherique.getAdresse());
00108 b.putInt("etat", Peripherique.CODE_DECONNEXION);
00109 b.putString("donnees", "");
00110 msg.setData(b);
00111 handler.sendMessage(msg);
00112 Log.d(TAG,"fin run() " + peripherique.getNom() + "[" + peripherique.getAdresse() + "]");
00113 }
00114
00115 public void arreter()
00120 {
00121     Log.d(TAG,"arreter() " + peripherique.getNom() + "[" + peripherique.getAdresse() + "]");
00122     if(!estFini)
00123     {
00124         estFini = true;
00125     }
00126     try
00127     {
00128         Thread.sleep(250);
00129     }
00130     catch (InterruptedException e)
00131     {
00132         e.printStackTrace();
00133     }
00134 }
00135 }

```

## 8.27 Référence du fichier TypeJeu.java

Déclaration de la classe TypeJeu.

### Classes

- class [projet.lasalle84.darts.TypeJeu](#)  
Déclaration de la classe *TypeJeu*.

### Paquetages

- package [projet.lasalle84.darts](#)

### 8.27.1 Description détaillée

Déclaration de la classe TypeJeu.

#### Auteur

Menella Erwan

Définition dans le fichier [TypeJeu.java](#).

## 8.28 TypeJeu.java

```
00001 package projet.lasalle84.darts;
00002
00009 import android.util.Log;
00010
00015 public class TypeJeu
00016 {
00020     private boolean doubleOut;
00021     private int pointDepart;
00022
00026     private final static int PARTIE_501 = 0;
00027     private final static int PARTIE_501_DOUBLE_OUT = 1;
00028     private final static int PARTIE_301 = 2;
00029     private final static int PARTIE_301_DOUBLE_OUT = 3;
00030     private final static String TAG = "TypeJeu";
00031
00038     public TypeJeu()
00039     {
00040         this.pointDepart = 501;
00041         this.doubleOut = true;
00042     }
00043
00051     public TypeJeu(int pointDepart, boolean doubleOut)
00052     {
00053         this.pointDepart = pointDepart;
00054         this.doubleOut = doubleOut;
00055     }
00056
00063     public TypeJeu(int idModeJeu)
00064     {
00065         switch (idModeJeu)
00066         {
00067             case PARTIE_501:
00068                 this.pointDepart = 501;
00069                 this.doubleOut = false;
00070                 break;
00071             case PARTIE_501_DOUBLE_OUT:
00072                 this.pointDepart = 501;
00073                 this.doubleOut = true;
00074                 break;
00075             case PARTIE_301:
00076                 this.pointDepart = 301;
00077                 this.doubleOut = false;
00078                 break;
00079             case PARTIE_301_DOUBLE_OUT:
00080                 this.pointDepart = 301;
00081                 this.doubleOut = true;
00082                 break;
00083         }
00084     }
00085 }
00086 public int getPointDepart ()
00087 {
00088     return pointDepart;
00089 }
00090
00091 public void setPointDepart(int pointDepart)
00092 {
00093     this.pointDepart = pointDepart;
00094 }
00095
00096 public boolean estDoubleOut ()
00097 {
00098     return doubleOut;
00099 }
00100
00101 public void setDoubleOut(boolean doubleOut)
00102 {
00103     this.doubleOut = doubleOut;
00104 }
```

```
00105
00106 public String getTypeJeu()
00107 {
00108     String typeJeu = Integer.toString(this.getPointDepart());
00109     if (doubleOut)
00110     {
00111         typeJeu = typeJeu.concat("_DOUBLE_OUT");
00112     }
00113     Log.d(TAG, "getTypeJeu: "+ typeJeu);
00114     return typeJeu;
00115 }
00116 }
00117 }
```