

M4104C – Projet Android

Mes Ratios – Compte rendu final

IUT du Limousin, département Informatique

DIM Gaël

MAUREL Guillaume

LAGOUTTE Pierre

Groupe G4



Université
de Limoges



INSTITUT UNIVERSITAIRE
DE TECHNOLOGIE



Introduction

Dans le cadre du module M4104C, nous devons créer une application mobile Android avec l'IDE Android Studio. Le choix du thème de l'application est libre, mais nous devons quand même développer une application avec des fonctionnalités suffisamment complexe pour que nous puissions apprendre le plus possible le développement Android Studio. Nous avons donc choisi de développer une application permettant d'enregistrer des ratios de victoire/défaite et de pouvoir les consulter par la suite. Ces ratios seront enregistrés en fonction de jeux. Chaque jeu sera enregistré par l'utilisateur. Ensuite l'utilisateur pourra visualiser ses résultats dans un planning. Il pourra aussi partager ses ratios avec ses amis. Par ailleurs l'utilisateur pourra aussi changer le thème de l'application.

Sommaire

Introduction.....	2
1 Rappel du sujet.....	4
1.1 Description fonctionnelle.....	4
1.2 Maquettes réalisées.....	5
2 Les choix techniques faits.....	8
2.1 La base de données.....	8
2.2 Importation d'image depuis la galerie.....	8
2.3 Partage de son ratio à ses amis.....	9
3 Les fonctionnalités implémentées.....	10
3.1 Affichage des jeux.....	10
3.2 Enregistrement des jeux.....	11
3.3 Enregistrement d'un ratio.....	12
3.3.1 Enregistrement basique d'un ratio.....	12
3.3.2 Reprise d'un ratio en cours.....	13
3.4 Affichage d'une fenêtre statistiques.....	14
3.4.1 Modification/ajout du commentaire.....	16
3.4.2 Partage de son ratio.....	16
4 Difficultés rencontrées.....	18
4.1 Erreur lors de sélections de certaines views.....	18
4.2 Problèmes lors de l'utilisation d'image.....	18
Conclusion.....	19

1 Rappel du sujet

1.1 Description fonctionnelle

Comme écrit dans l'introduction, l'application que nous devons développer permettra d'enregistrer des ratios. Et ces ratios seront reliés à un seul jeu créé par l'utilisateur. Chaque ratio sera ensuite consultable.

D'après le cahier des charges, lorsque l'application sera lancée, une page listant tous les jeux sera affichée. Chaque jeu sera accompagné d'une image, de son nom, et d'un bouton permettant de le supprimer. Par ailleurs dans cette page il y aura un bouton pour ajouter un jeu. L'utilisateur devra alors préciser le nom du jeu et choisir une image. L'utilisateur devra choisir une des images de la galerie de son appareil Android. Il ne lui restera plus qu'à valider ces informations avec le bouton Valider.

Lorsqu'on clique sur un jeu, une page s'ouvre proposant deux options : « Enregistrer un nouveau ratio », ou « Consulter ses anciens ratios ».

Lors de l'enregistrement d'un ratio, on pourra incrémenter le nombre de victoires et le nombres de défaites. On peut aussi décrémenter ces derniers. Nous avons aussi la possibilité d'ajouter un commentaire à ce ratio. Ce commentaire sera consultable par la suite dans les statistiques. Dans ce commentaire on peut d'ailleurs ajouter une photo, soit de la galerie soit de l'appareil photo.

La consultation de nos anciens ratios ouvre une page affichant un planning. Sur ce planning est affiché le ratio de victoire/défaite par jour du jeu sélectionné. Si le cadre est vert, alors le ratio victoire/défaite est positif, sinon, il est rouge. L'utilisateur peut choisir le jour où les ratios doivent être affichés. En cliquant sur un cadre, l'utilisateur peut voir le commentaire qu'il a rentré s'afficher.

1. 2 Maquettes réalisées



Maquette - Écran d'accueil

Voici l'écran d'accueil sur lequel l'utilisateur va arriver lorsqu'il lancera l'application.



Maquette – Après avoir sélectionné un jeu

Après avoir sélectionné un jeu, l'utilisateur arrive ici.



Maquette – Statistiques journalières

Après avoir cliqué sur « CONSULTER RATIOS », cet écran s'affiche avec chaque ratio classé par ordre chronologique en fonction de l'heure. Le jour affiché par défaut est aujourd'hui. L'utilisateur peut choisir le jour dans la toolbar au dessus.



Maquette – Nouveau Ratio

Voici l'écran affiché après que l'utilisateur choisisse de créer un nouveau ratio. Il est possible d'incrémenter et de décrémenter le ratio. Le nombre total de partie est affiché au milieu. Le commentaire n'est pas affiché dans cette maquette car nous avons décidé de rajouter cette fonctionnalité après avoir créé cette maquette. Le nombre de victoire et de défaite ne peuvent pas excéder 99 et ne peuvent pas être inférieurs à 0.

2 Les choix techniques faits

2.1 La base de données

Pour réaliser cette application, nous devons créer une base de données comportant 3 tables : Ratio, Jeu et Commentaire. Mais lors du développement, nous avons constaté que nous n'aurons pas le temps d'implémenter l'image dans un commentaire. Nous avons donc décidé de fonctionner avec 2 tables : Ratio et Jeu. Le commentaire sera seulement une chaîne de caractères contenu dans la table Ratio. La particularité de ces tables et qu'elles sont reliées par une foreign key. Ratio contient une foreign key vers la table Jeu.

```
public static final String TABLE_CREATE = "create table "+ TABLE_NOM +" ("
+COLONNE_ID+" integer primary key autoincrement,"
+COLONNE_DATE+" text not null,"
+COLONNE_NB_VICTOIRES+" integer,"
+COLONNE_NB_DEFAITES+" integer,"
+COLONNE_ID_JEU + " integer,"
+COLONNE_COMMENTAIRE+ " text,"
+COLONNE_EN_COURS + " number"
+", FOREIGN KEY (" +COLONNE_ID_JEU+" ) references "+JeuDataSource.TABLE_JEU+" (" +JeuDataSource.COLONNE_ID+" )"
+");";
```

Code RatioDataSource.java - Script de création de la table Ratio

Par ailleurs, lors de la suppression d'un jeu, nous effaçons en même temps tous les ratios associés. Cette foreign key permet de garder une base de données cohérente.

2.2 Importation d'image depuis la galerie

Notre projet contient plusieurs utilisations d'images. Nous devons donc comprendre comment laisser l'utilisateur choisir une image, puis comprendre comment la sauvegarder et la réafficher dans une ImageView.

La sélection d'image est très simple, elle est gérée par Android. Il suffit de lancer un Intent en précisant qu'on souhaite récupérer une image.

```
case R.id.image :
    Intent gallery = new Intent(Intent.ACTION_PICK,
    android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    gallery.setType("image/*");
    startActivityForResult(gallery, RESULT_LOAD_IMAGE);
```

AjouterJeuActivity.java – Intent pour récupérer une image

Après l'exécution de ces lignes, l'application va ouvrir l'application Galerie Photos de l'appareil Android. L'utilisateur pourra ensuite choisir sa photo en cliquant dessus.

Ensuite, l'application va récupérer l'« URI » de l'image. L'URI peut ressembler au chemin absolu de l'image. C'est ce qui va permettre de récupérer et d'afficher l'image, c'est une

propriété de l'imageView. Pour récupérer cet URI, l'application doit implémenter la méthode onActivityResult qui va retourner l'URI sélectionnée par l'utilisateur.

Cette Uri sera ensuite stocker dans la table Jeu sous forme d'une chaine de caractères. Pour la réutiliser, il suffit de récupérer l'image view sur laquelle on veut mettre l'image, puis lui appliquer la méthode setImageUri() avec l'URI en paramètre.

2.3 Partage de son ratio à ses amis

Au début, nous pensions implémenter une liste d'amis avec un bouton « Envoyer Ratio » qui enverrait un sms à un ami. Mais nous avons ensuite décider qu'ils seraient plus intéressant de passer par une des fonctionnalités natives, en effet Android propose un system d'Intent qui permet de faire des demandes à d'autres applications.

```
private void partager(Ratio ratio, String nomJeu) {  
    Intent intentShare = new Intent(Intent.ACTION_SEND);  
    intentShare.setType("text/plain");  
    intentShare.putExtra(Intent.EXTRA_TEXT, "Dans mes dernières game de "+nomJeu+" j'ai fait "+ratio.getNbVictoire  
    context.startActivity(Intent.createChooser(intentShare, "Partager son ratio avec "));  
}
```

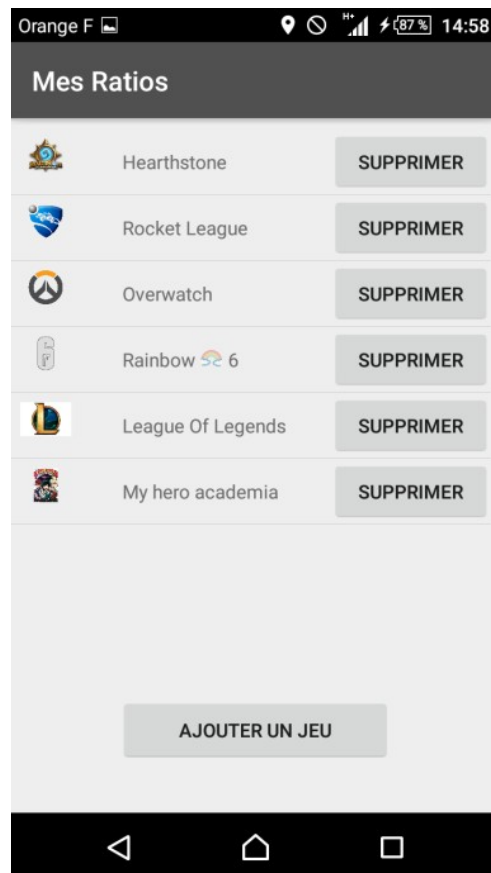
Code – Partage de Ratio via un Intent

Quand Android va exécuter ces lignes, une liste d'applications va s'afficher. Ce sont les applications qui peuvent traiter l'Intent envoyé (de type ACTION_SEND). Grâce à cela, le partage de ratio peut passer par l'application messagerie du téléphone (SMS), mais aussi par Twitter, Facebook, Messenger, ou toutes autre applications pouvant envoyer un message.

3 Les fonctionnalités implémentées

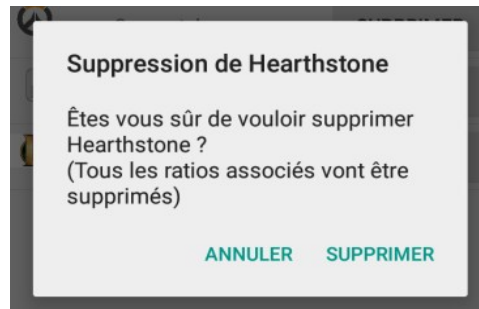
Il nous manque beaucoup de fonctionnalités tel que le planning ou l'insertion d'image dans les commentaires. Mais toutes les autres fonctionnalités sont là et au fur et à mesure du développement nous nous sommes rendus que certaines fonctionnalités seraient intéressantes à rajouter. Voici toutes les fonctionnalités du projet et comment y accéder.

3.1 Affichage des jeux



Screenshot – Menu Principal

La liste des jeux enregistré a été implémenté, il est possible de cliquer sur ces jeux, pour accéder au menu du jeu. Chaque jeu listé possède aussi un bouton supprimer qui permet de supprimer le jeu. Mais un message d'avertissement apparaît car supprimer un jeu supprime aussi tous les ratios associés.

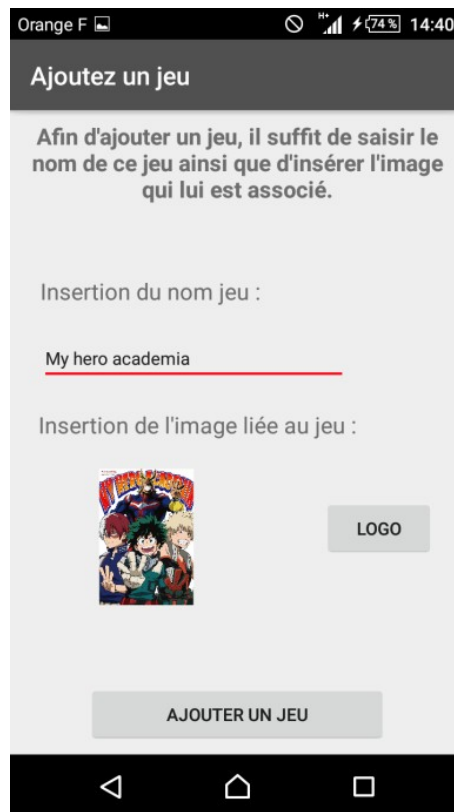


Screenshot – Message Avertissement lors de la suppression d'un jeu

3.2 Enregistrement des jeux

Le bouton ajouter un jeu dans le menu principal permet de rajouter un jeu en lui spécifiant un nom et une image. Si une des informations n'a pas été rempli, un message d'erreur apparaît sous la forme d'un Toast.

Lorsqu'on clique sur Logo, l'application affiche la Galerie du téléphone. Il est ensuite possible de définir quel image on souhaite utiliser pour ce jeu.

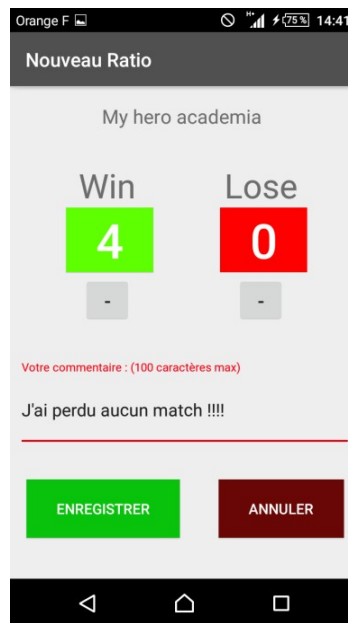


Screenshot – Insertion d'un jeu

3.3 Enregistrement d'un ratio

3.3.1 Enregistrement basique d'un ratio

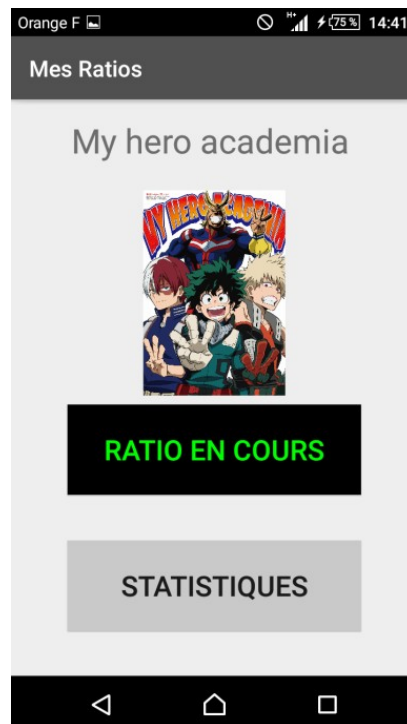
On peut incrémenter et décrémenter le nombre de victoires et de défaites. Le nombre de victoires et de défaites ne peuvent pas être supérieur à 99 et inférieurs à 0. Si l'utilisateur essaie de les dépasser, il ne se passe rien. Par ailleurs l'utilisateur peut insérer un commentaire à son ratio. Pour valider le ratio, il suffit d'appuyer sur le bouton « Enregistrer ».



Screenshot – Enregistrement d'un ratio

3.3.2 Reprise d'un ratio en cours

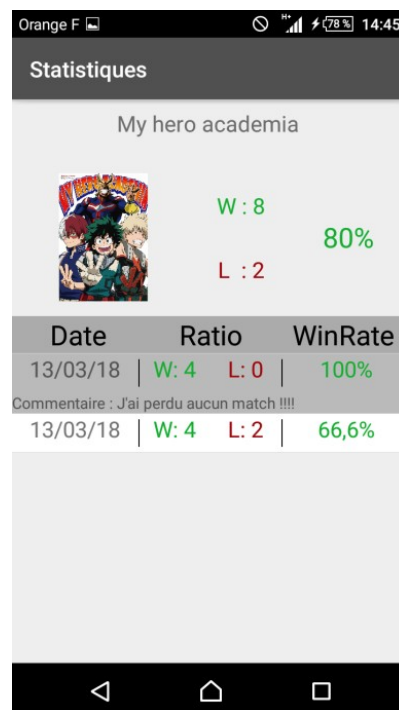
Une des fonctionnalités que nous avons rajouté est la possibilité de ne pas perdre son ratio en pleine modification si on quitte l'application. Nous avons rajouté la méthode `estEnCours()` dans l'objet `Ratio`. Si la base de données trouve qu'un ratio est en cours, alors lors de la sélection du jeu il affiche Ratio en cours au lieu de Nouveau Ratio. L'utilisateur peut supprimer ce ratio en cours en appuyant sur le bouton Annuler.



Screenshot – Ratio en cours

3.4 Affichage d'une fenêtre statistiques

Faute de temps, nous n'avons pas pu implémenter la fonctionnalité du planning lors de la consultation de ses anciens ratios. Mais au lieu de cela nous avons fait un affichage relativement simple.

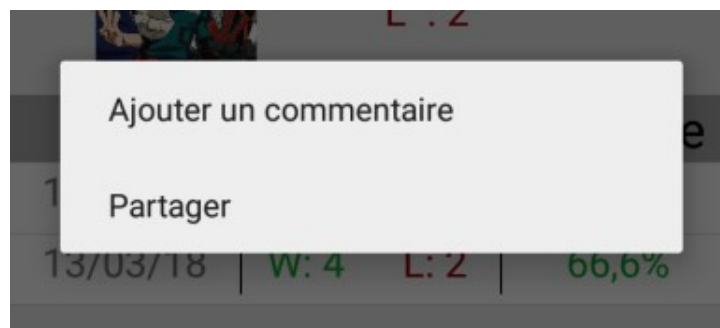


Screenshot – Page Statistiques

La page statistiques rappelle le titre et l'image du jeu. Elle affiche le nombre total de victoires et de défaites des ratios enregistré avec ce jeu. Elle affiche aussi le pourcentage global de ces ratios. Le pourcentage est affiché en vert s'il est supérieur ou égal à 50 %, sinon il est affiché en rouge.

En dessous chaque ratio est affiché par ordre chronologique. Il y est affiché la date, le nombre de victoire et de défaites, puis un pourcentage de ce ratio est affiché. Il est affiché soit en vert soit en rouge en fonction de la valeur du pourcentage.

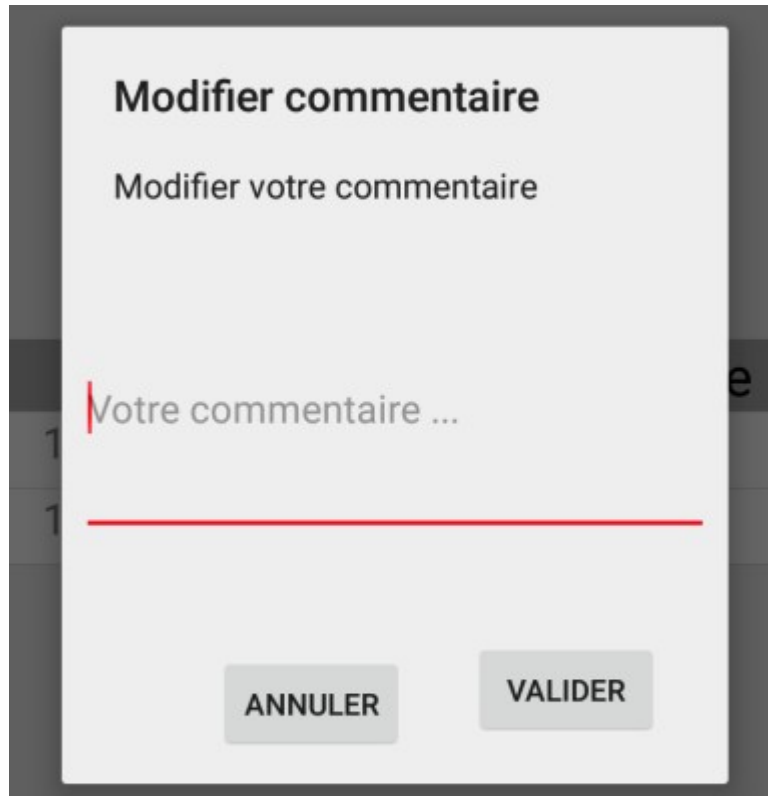
Si on appuie une fois sur le ratio, cela affiche le commentaire associé au ratio s'il en possède un. Puis si on appuie longuement sur le ratio. Deux options s'offre à nous : l'ajout/modification du commentaire, ou partager le ratio.



Screenshot – Appui long ratio

3.4.1 Modification/ajout du commentaire

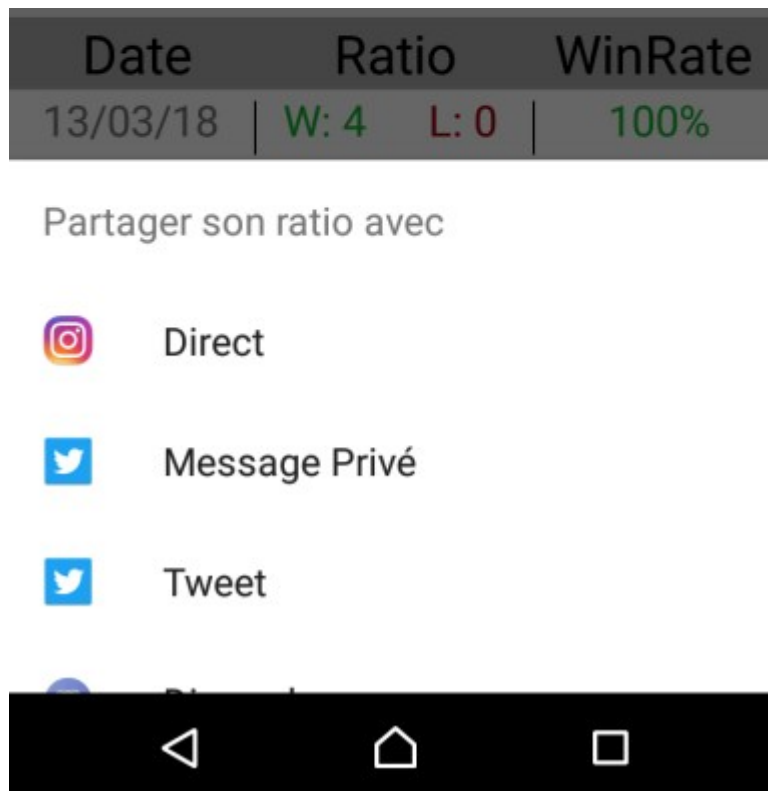
Si le ratio n'a pas de commentaire, l'option s'appelle « Ajouter un commentaire », sinon elle s'appelle « Modifier Commentaire ». Lorsqu'on la choisit, une autre boîte de dialogue s'ouvre avec un champ de saisie et deux boutons. Un bouton pour valider et un bouton pour Annuler.



Screenshot – Ajout d'un commentaire

3.4.2 Partage de son ratio

Lorsque l'on choisit l'option Partager, une liste d'application s'ouvre. L'utilisateur n'a plus qu'à choisir laquelle il souhaite utiliser pour partager son ratio.



Screenshot – Liste des applications de partage

Si on choisit l'application Messagerie, par exemple, un message déjà pré-rédigé est écrit dans la zone de saisie de texte. L'utilisateur peut le modifier s'il le souhaite. Puis il n'a plus qu'à paramétrer son message puis à l'envoyer.

4 Difficultés rencontrées

4.1 Erreur lors de sélections de certaines views

Android retournait un bug lorsque nous essayons de sélectionner une view avec la méthode `findViewById()`. L'erreur nous indiquait qu'aucune view ne correspondait à l'id 0x0000. Cette erreur nous est arrivé plusieurs fois.

Elle nous est arrivé lors du développement de la page Statistiques. La ligne au dessus de la liste des ratios est un `LinearLayout`, elle contient l'image du jeu, le nombre de victoires/défaites, et le pourcentage global. Le nombre de victoires et de défaites étaient dans un sous-`GridLayout`. On ne pouvait pas sélectionner ces deux `TextView`. On ne trouvait aucune réponse sur Internet malgré tout le temps que nous y avons passé, nous avons donc outrepassé le problème en enlevant ces `TextView` du `GridLayout`, et en les affichant dans un `LinearLayout` de type vertical. Après avoir fait cette modification, la selection des Views fonctionnait.

Cette erreur nous est arrivé lors de la création de la boite de dialogue pour la modification du commentaire. En effet, normalement une boîte de dialogue possède les méthodes `setPositiveButton` et `setNegativeButton`. Mais cette fois ci nous avons outrepassé le problème sans beaucoup chercher sur Internet. Nous avons rajouté deux bouton dans l'alert Dialog et avons gérer les actions associés à ces derniers nous-mêmes. Dans le bouton annuler et valider, nous devions donc fermer la boite de dialogue. Pour cela nous avons récupérer l'alert dialog dans un objet et nous avons utilisé ses méthodes.

4.2 Problèmes lors de l'utilisation d'image

Nous avons aussi eu des problèmes d'affichage d'image. Lorsque l'on créé un nouveau jeu. Seul l'image de ce nouveau jeu était affiché dans le menu principal. Les autres disparaissait. Si on revenait sur l'application après l'avoir mise en arrière plan, aucune image n'apparaissait. Après de longue recherche, cela était dû au fait que nous n'avions pas mis de type à l'intent.

```
case R.id.image :  
    Intent gallery = new Intent(Intent.ACTION_PICK,  
                                android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);  
    gallery.setType("image/*");  
    startActivityForResult(gallery, RESULT_LOAD_IMAGE);
```

Nous lui avons donc rajouter le type « image/ », ce qui va dire à l'application de ne prendre que les images de la gallerie. Les images qu'on prenait ne faisait pas parti de la gallerie, nous n'avions pas les droits pour afficher ces derniers. Une autre solution aurait été de demander ces droits d'affichage

Conclusion

Beaucoup de fonctionnalités sont manquantes pour faute de temps et d'imprévus. Les erreurs que nous avons rencontrées nous ont pris beaucoup de temps. Mais nous avons quand même réalisé une application fonctionnelle et complète. Elle est prête à utilisation. Nous avons cependant quelques dernières choses à corriger comme des erreurs d'affichage. Nous pouvons aussi ajouter certaines fonctionnalités manquantes comme par exemple la personnalisation d'un jeu, avoir la possibilité de changer son nom et son image. Puis pour permettre ces modifications nous devons revoir l'affichage dans le menu principal, c'est à dire modifier le bouton supprimer pour le mettre ailleurs à côté du bouton modifier. Aussi notre application manque d'une personnalité graphique, son design est joli mais n'est pas très original