

Rapport - Développement Mobile Démineur

Kamarouzamane Combo, Jeremie Legros
L3 informatique

29 avril 2018

Résumé

Dans ce rapport, nous allons vous présenter ce que nous avons pu voir dans l'UE Développement pour Mobile, et le projet que l'on a fait cette année. Nous avons choisi de nous inspirer du jeu Démineur et d'en faire une version Android et iOS. Nous vous exposons ici le principe du jeu, le fonctionnement de l'application et son développement.

Mots clés : Android Studio, Xcode, Java, Swift, Activity

1 Introduction

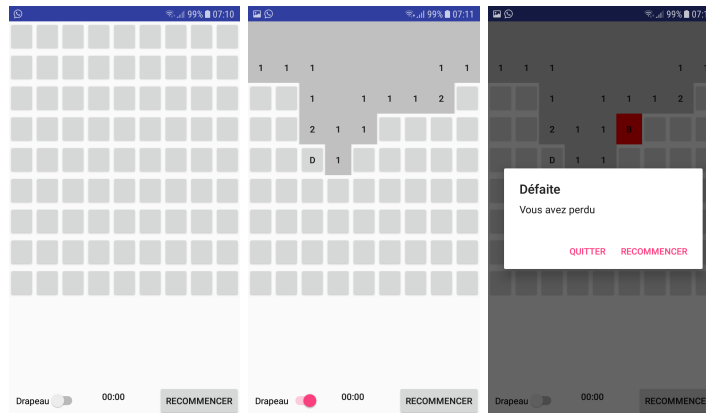
Dans le cadre de l'UE Développement pour Mobile, nous avons décidé de réaliser une version simple du jeu Démineur. Il nous était demandé de réaliser le même jeu sur les deux plateformes (Android et iOS. [3]). La version Android est programmée en Java et celle iOS est programmée en Swift. Pour commencer, nous vous présenterons d'abord quel type de jeu nous avons voulu développer, puis nous explorerons l'architecture du code.

2 Description générale de l'application

Comment jouer au Démineur ?

Vous devez déminer ce champ de mines. Pour cela, il a été découpé en carrés, qui peuvent contenir soit une seule mine Mine Démineur, soit aucune. Au début, les mines, enterrées, ne sont évidemment pas visibles. Pour gagner au Démineur, il vous faut déterminer l'emplacement de toutes les mines. Pour atteindre cet objectif, vous pourrez prendre connaissance du nombre de mines (**1,2,3 ou 4**) dans le voisinage immédiat d'une case, en appuyant sur celle-ci et à condition qu'elle soit vide. Si jamais vous appuyez sur une case contenant une mine, celle-ci explose (**B**) Explosion Démineur et vous perdez. Il est possible de planter un drapeau (en utilisant le **bouton switch**) pour marquer l'emplacement supposé d'une mine. Si la case appuyée est vide, toutes les cases vides adjacentes s'ouvriront récursivement.

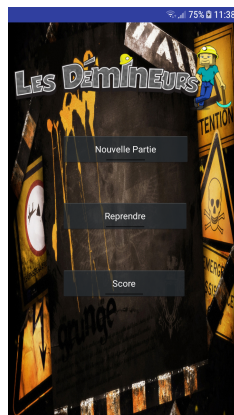
Pour gagner, il ne doit pas y avoir de mines incorrectement marquées avec un drapeau. Par contre, il n'est pas nécessaire de marquer toutes les mines pour terminer ; vous avez juste besoin de déterminer toutes les cases vides.



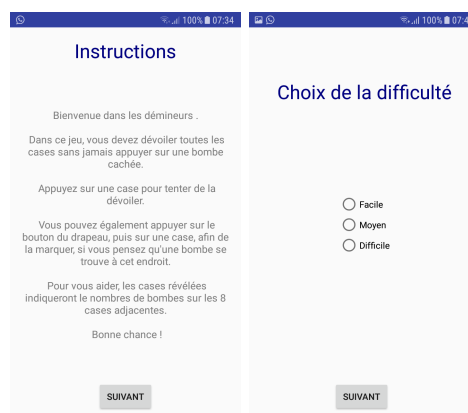
Mais que fait-elle exactement ?

Au lancement de l'application, la page d'accueil apparaît et donne la possibilité à l'utilisateur de jouer une nouvelle partie (choix : le niveau de difficulté), de reprendre une partie sauvegardée ou de consulter la liste des scores enregistrés.

Voici une capture d'écran du Main Activity :



Avant que l'utilisateur puisse choisir le niveau de difficulté, on a mis en place une activity de présentation du jeu et des règles du jeu. En suite il saisir la difficulté, toute fois s'il veut accéder au jeu sans avoir saisi de difficulté au préalable, il s'affichera un message d'erreur qui lui demandera de choisir un niveau de difficulté.



L'application est jouable sur *Android* [1] et *iOS* [2], elle fonctionne correctement sur tout type d'écran (grand, petit, ...), en mode portrait et paysage.

3 Architecture du code

3.1 Accueil et activityaccueil.xml

L'écran d'accueil affiche le titre du jeu et comporte trois boutons, l'un pour lancer une nouvelle partie et le second pour accéder à la liste des parties en cours enregistrées, et la dernière pour accéder à la liste des scores enregistrés. Lorsqu'on appuie sur le premier, on est dirigé vers l'activité d'introduction, tandis que les autres nous envoient sur des activités gérant les parties en cours et les scores.

3.2 Android

La classe la plus importante est `GameActivity`, elle va générer la grille grâce aux informations saisies au préalable. On peut remplir le plateau à partir de rien à l'aide de la fonction `fillTableau()` ou bien à partir d'une sauvegarde antérieure avec la fonction `fillTableau(Cellule[][] plateau)`.

```
/**
 * Remplit le plateau de jeu à partir de rien
 */

private void fillTableau(){
    plateau = new Cellule[taille][taille];
    // On remplit le plateau
    for (int y = 0; y < taille; y++) {
        TableRow r = new TableRow(this);
        table.addView(r);
        for (int x = 0; x < taille; x++) {
            Cellule c = new Cellule(this,x,y);
            c.setOnClickListener(onClickCellule);
            r.addView(c);
            plateau[x][y] = c;
        }
    }
}

/**
 * Remplit le tableau de jeu à partir d'une sauvegarde antérieure
 * @param plateau le plateau à réutiliser
 */

private void fillTableau(Cellule[][] plateau){
    for(int y = 0; y < taille; y++){
        TableRow r = new TableRow(this);
        table.addView(r);
        for (int x = 0; x < taille; x++) {
            Cellule c = plateau[x][y];

            ((ViewGroup)c.getParent()).removeView(c);
            if(!c.isRevele())
                c.setOnClickListener(onClickCellule);
            r.addView(c);
        }
    }
}
```

```
    }
}
```

Pour la gestion de mines/bombe, elle se fait de façon aléatoire

```
/**
 * Place les bombes sur le plateau
 */
private void addBombes(){
    int cptBombe = bombe;
    while (cptBombe>0){
        Random gen = new Random();
        int x = gen.nextInt(taille);
        int y = gen.nextInt(taille);

        if(!plateau[x][y].isBombe()){
            plateau[x][y].devenirBombe();
            cptBombe--;
        }
    }
}
```

Pour vérifier que le joueur à bien *gagner*, on parcourt tout le tableau et on regarde que toutes les case ont étaient révélé et que les bombes ont étaient trouvé.

```
/**
 * Vérifie si la partie est terminée
 */
private void testVictoire(){
    for(int i=0; i<taille; i++) {
        for (int j = 0; j < taille; j++) {
            if ((!plateau[i][j].isRevele()) && (!plateau[i][j].isBombe())) {
                return;
            }
        }
    }
}
victoire();
```

•

4 Conclusion générale

Cette projet était une bonne expérience, et qui plus était enrichissante. Grâce elle, elle nous a permis de pouvoir développer des applications sur les deux système, Android et iOS, de créer plusieurs projets,cette UE nous permis de pouvoir travailler en autonomie et de ce fait, effectuer nos propres recherches, tester de nouvelles choses, faire des erreurs, et en apprendre davantage grâce à elles.

Références

- [1] Android developers. <https://developer.android.com>.
- [2] iOS developer. <https://developer.apple.com>.
- [3] Système d'exploitation mobile. https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation_mobile.