

# DE Développement de systèmes mobiles et sans fil 2014-2015

## Document:

2 feuilles A4 recto-verso manuscrites (non imprimées, non photocopiées, ...) sont autorisées. Tout autre document est interdit. Les calculatrices et les téléphones portables sont interdits. Le barème vous est donné à titre indicatif.

## Conseil:

Commencer par lire l'ensemble de l'énoncé avant de commencer à répondre aux questions. Même si vous ne savez pas répondre à la question en entier, écrivez ce que vous savez. Dans chacune des cases réponses, vous avez largement la place pour répondre. Cependant, si vous manquez de place, la dernière page est à votre disposition. Le barème est donné à titre indicatif, il comprend des points de bonus.

## **Exercice 1) Questions de Cours**

**Question 1** - Dans le cadre de la programmation mobile, définir la compilation croisée. Donnez un exemple concret. Quels sont les avantages et les inconvénients de cette approche ?

Compilation native -> Exécution dans un simulateur

Compilation croisée -> Transfert de l'application mobile sur le téléphone

On parle de compilation croisée lorsque le système sur lequel tourne le compilateur n'est pas le même que celui pour lequel sont produits les binaires. (source : <http://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-112/Les-dessous-d-Android>)

**Question 2** - Quel est le cycle de vie d'une application Android ?

Schéma du cycle de vie d'une application android:

<http://developer.android.com/guide/topics/processes/process-lifecycle.html>

**Question 3** - Que trouve-t-on dans le fichier `res/values/string.xml` ?

Ensemble des attributs de type string utilisés dans l'application (source : <http://developer.android.com/guide/topics/resources/string-resource.html>)

## Question 4 - Comment intégrer Google Map à une application ?

Description de l'inclusion de Google Map au sein de l'application, droits d'utilisations, données utilisées par le module, cadre de fonctionnement, etc.

Utilisation de l'API Google Maps ?

## Question 5 - D'après votre expérience, quelles sont les difficultés que l'on peut rencontrer lorsque l'on fait du développement natif pour Android ?

En terme de mise en pratique il faut s'équiper d'un téléphone android pour tester l'application car les émulateurs ne sont pas assez optimisés et fluides pour faire ces tests.

La programmation d'interface en Java peut être assez lourde, c'est pourquoi on utilise des templates XML qui nous permettent de simplifier la programmation de des vues, la mise en place et les modifications se retrouves très simplifier et donc on obtient plus de flexibilité pour la modification du code quant à l'évolution de l'application.

## Question 6 - Donner le code Java de l'interface principale de votre jeu (celle où le tableau de jeu apparaît) ?

Code de l'interface principale:

## Exercice 2) Le Tic-tac-toe

*Le Tic-tac-toe se joue sur une grille carrée de 3x3 cases. Deux joueurs s'affrontent. Ils doivent remplir chacun à leur tour une case de la grille avec le symbole qui leur est attribué: O ou X. Le gagnant est celui qui arrive à aligner trois symboles identiques, horizontalement, verticalement ou en diagonale. Dans cette version du jeu, l'utilisateur (le joueur) va jouer contre l'ordinateur.*

## Question 1 - Dessiner et commenter toutes les différentes interfaces types de cette application. Nommer chacune d'entre elles.

- Ecran d'accueil
- Options (Reset Stats, Audio On-Off, Music On-Off)
- Scores - Statistiques de jeux
- Interface début de partie (choix de la difficulté)
- Interface de Jeux - Partie

## Question 2 - Définir le manifest de cette application.

Voir le code pour le manifest.

**Question 3** - Donner le code permettant de définir l'une de ces interfaces. Bien identifier quelle interface vous avez programmé.

Code définissant une interface:  
(XML et Java)

**Question 4** - Donner le code permettant de programmer le modèle de votre jeu.

```
class Modele {
    private int list [width*height];          [0] [1] [2] // Game table    // Init list = -1 for all;
    private int width;                        [3] [4] [5]
    private int height;                       [6] [7] [8]    // Include des Setters - Getters

    public Modele(int widthAndHeight) {
        setWidth(widthAndHeight);    // We can use only one value instead of 2
        setHeight(widthAndHeight);   // But the code is more explicit with 2
    }

    public boolean play(int case, int elem){
        // Check if 0 < case < (width*height -1) // Check if elem == 1 or 0
        if( list[case] == -1) { list[case] = elem; return true; } else { return false; }
    }

    public boolean checkWin(int elem) {      // Check if elem == 1 or 0
        int validationW;
        int validationH [width]            = 0 for all;
        int validationDiagHL = 0;
        int validationDiagHR = 0;

        for(int i = 0; i < height; i++) {
            validationW = 0;
            for(int j = 0; j < width; j++) {
                if( list[ j + i * width ] == elem) {
                    validationW++;
                    validationH[ j ]++;
                }
            }
            // 1 case to check for Diag
            if( list[ i * (width + 1) ] == elem) { validationDiagHL++; }
            if( list[ i + ( i + 1 ) * (width - 1) ] == elem){ validationDiagHR++; }

            if(validationW == width){ return true; }
        }
    }
}
```

```
for(int cpt; cpt < height; cpt ++){  
    if(validationH[cpt] == height) {                return true; }  
    }  
    if( validationDiagHL == width || validationDiagHR == width) { return true; }  
    }  
}
```

**Question 5** - Comment faites vous le lien entre le modèle que vous venez de définir et les interfaces que vous avez nommées à la question 1 ?