

## PROJET LUDO



Les missions que j'ai réalisées, dans le cadre des Travaux Pratiques en cours de la formation à partir de l'application Android Studio, sont les suivantes :

- Développer un jeu de bataille,
- Créer une application permettant à un enfant de s'entraîner à compter et à calculer.

Android Studio est un environnement de développement permettant de développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il permet de créer des applications pour smartphones, tablettes tactiles, PDA, smartwatches (version Wear) et terminaux mobiles.

Android Studio est l'environnement de développement intégré de la plate-forme Android de Google. Les versions d'Android Studio sont compatibles avec certains systèmes d'exploitation Apple, Windows et Linux. Avec la prise en charge de Google Cloud Platform et de l'intégration d'applications Google, Android Studio offre aux développeurs une boîte à outils bien fournie pour créer des applications Android ou d'autres projets, et fait partie intégrante du développement Android depuis 2013.

Android Studio permet d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose des outils pour gérer le développement d'applications utilisant plusieurs langages et permet de visualiser rapidement la mise en page des écrans avec des résolutions variées simultanément à l'aide d'un émulateur faisant tourner un système Android virtuel sur un ordinateur.

Langages utilisés par Android studio : Java, Kotlin, JavaScript, C++.

Les prérequis techniques pour l'installation d'Android Studio : Un ordinateur exécutant une version 64 bits de Windows (8, 10 ou 11), Linux, macOS (10.14 Mojave ou version ultérieure) ou Chrome OS.

## JEU DE BATAILLE

### **Contexte et définition du besoin :**

L'association d'éducation populaire, MJC AGORA, dont le but est de favoriser l'épanouissement de la personne par l'accès à l'éducation et à la culture pour tous, souhaite proposer à ses adhérents des jeux éducatifs qui seront utilisés sur des mobiles Android.

Elle ne dispose pas de service informatique. Ma mission consiste à développer un jeu de bataille accessible par les utilisateurs sur un mobile Android.

### **Description du projet**

J'ai développé un jeu de bataille. L'application Ludo entraîne les enfants à différencier des cartes. A chaque étape du jeu, il faut déterminer le joueur qui l'emporte en comparant les points attribués aux cartes, avec un nombre de points différent pour la couleur atout.

### **Contraintes**

L'application a été adaptée visuellement pour des enfants. Elle s'exécute sur les mobiles Android.

## Développement

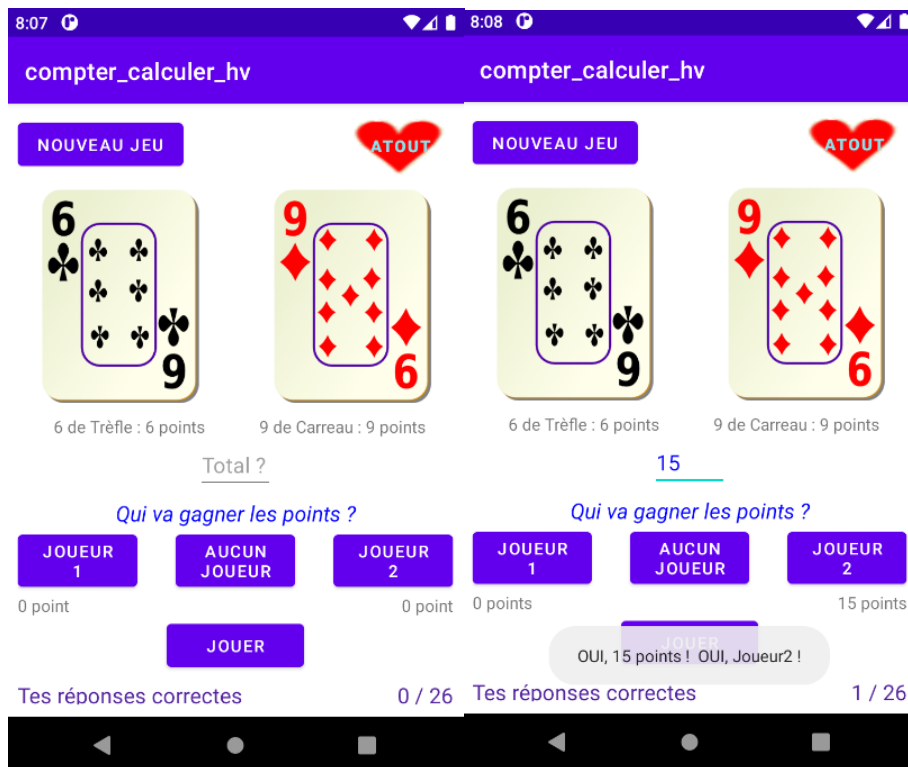
### Code permettant de distribuer les cartes

```
// mélanger le jeu de cartes
Carte carteTempo;
Random leHasard = new Random(); // générateur de nombres pseudo-aléatoires
for (int i = 0; i < tbJeu.length; i++) {
    // on tire une position au hasard entre 0 et
    // la méthode Next retourne un nombre aléatoire non négatif, inférieur au nombre maximal spécifié
    int indHasard = leHasard.nextInt( bound: tbJeu.length - 1);
    // on échange la ième carte avec celle à la position tirée au hasard
    carteTempo = tbJeu[indHasard];
    tbJeu[indHasard] = tbJeu[i];
    tbJeu[i] = carteTempo;
}

// distribuer le jeu à 2 joueurs
int indJoueur = 0;
for (int i = 0; i < tbJeu.length; i += 2) {
    tbJoueurUn[indJoueur] = tbJeu[i];
    tbJoueurDeux[indJoueur] = tbJeu[i + 1];
    indJoueur++;
}
```

### Interface obtenue





## JEU COMPTE-CALCULER

### Contexte et définition du besoin

La MJC AGORA, souhaite accompagner les jeunes enfants dans l'apprentissage du calcul et envisage de mettre à disposition de ceux-ci une application leur permettant de s'entraîner à compter et à calculer.

Ma mission consiste à développer une application « compter -Calculer » accessible sur un mobile Android.

### Description du projet

L'application compter-calculer est un ensemble de trois jeux dont le but est de dénombrer des lettres ou des images, ou de calculer des sommes.

### Contraintes

L'application a été adaptée visuellement pour des enfants. Elle s'exécute sur les mobiles Android.

### Développement

Le premier développement réalisé permet de présenter une suite de lettres choisies aléatoirement et demander de compter une lettre prise au hasard.

L'application génère une suite de 50 lettres tirées aléatoirement, et sélectionne aléatoirement une lettre à compter dans la liste.

Après avoir saisi une réponse dans la case total et validé, l'application affiche un message adapté.

### Code générant les lettres

```

4 usages
int compteur;
2 usages
char[] alphabet = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'};

```

```

Random random = new Random();
int lettreACompter = random.nextInt( bound: 26);
String consigne = "Compter le nombre de " + alphabet[lettreACompter];

compteur = 0;
String lettres = "";
for (int i = 0; i < 50; i++) {
    int position = random.nextInt( bound: 26);
    if (position == lettreACompter) compteur++;
    lettres += alphabet[position] + " ";
}

```

### Interface obtenue



Le second présente une suite de nombres au hasard et demande le total  
 L'application génère une suite de 5 nombres entre 0 et 10. Il faut ensuite calculer la somme de ces nombres et noter le résultat dans la case total.

### Code générant les nombres

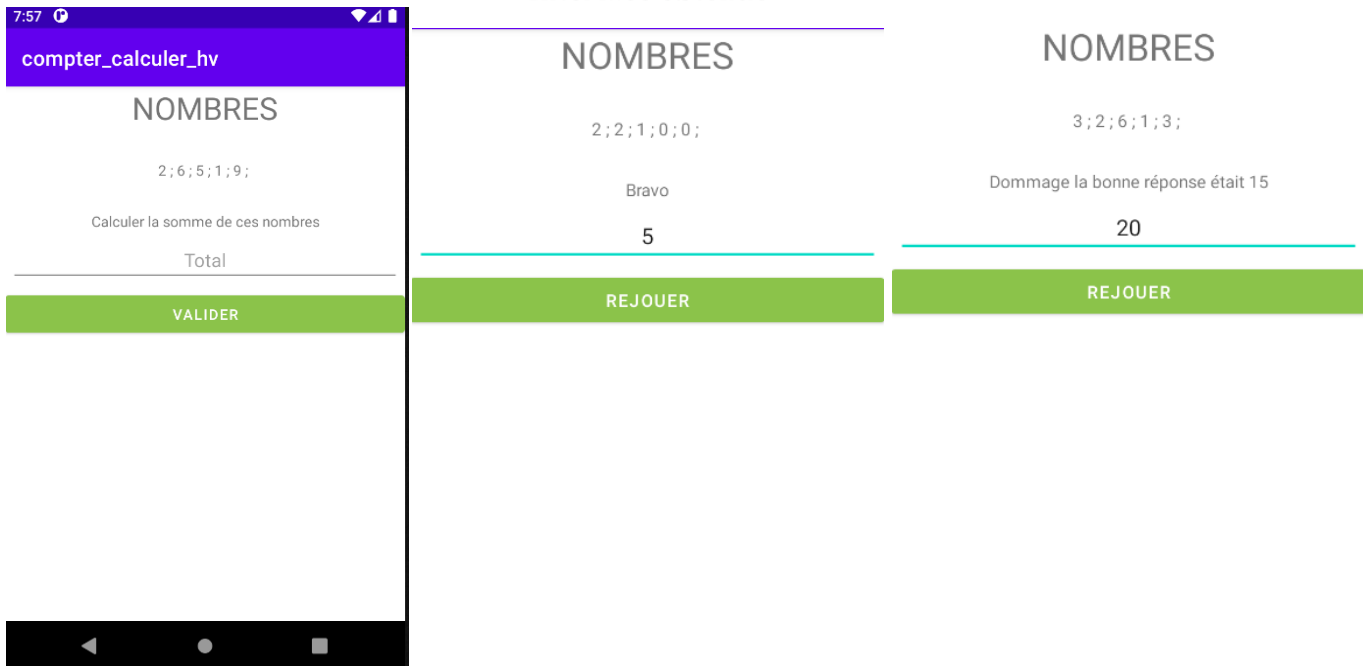
```

Random random = new Random();
String consigne = "Calculer la somme de ces nombres";

compteur = 0;
String nombres = "";
for (int i = 0; i < 5; i++) {
    int nombre = random.nextInt( bound: 11);
    compteur += nombre;
    nombres += nombre + " ";
}

```

## Interface obtenue



Le troisième présente une image sur laquelle il faut dénombrer des animaux.

L'application choisit aléatoirement une image dans une liste prédéfinie. Pour ces images, le décompte de chaque animal est connu par l'application. Elle choisit aléatoirement un animal dont on doit compter le nombre.

## Code pour le tirage aléatoire de l'image et de l'animal à dénombrer

```
2 usages
String[] animaux_a = {"chiens", "chats", "hiboux", "autruches", "tigres", "dromadaines", "toucans"};
1 usage
int[] nombres_a = {4, 2, 3, 6, 3, 3, 1};

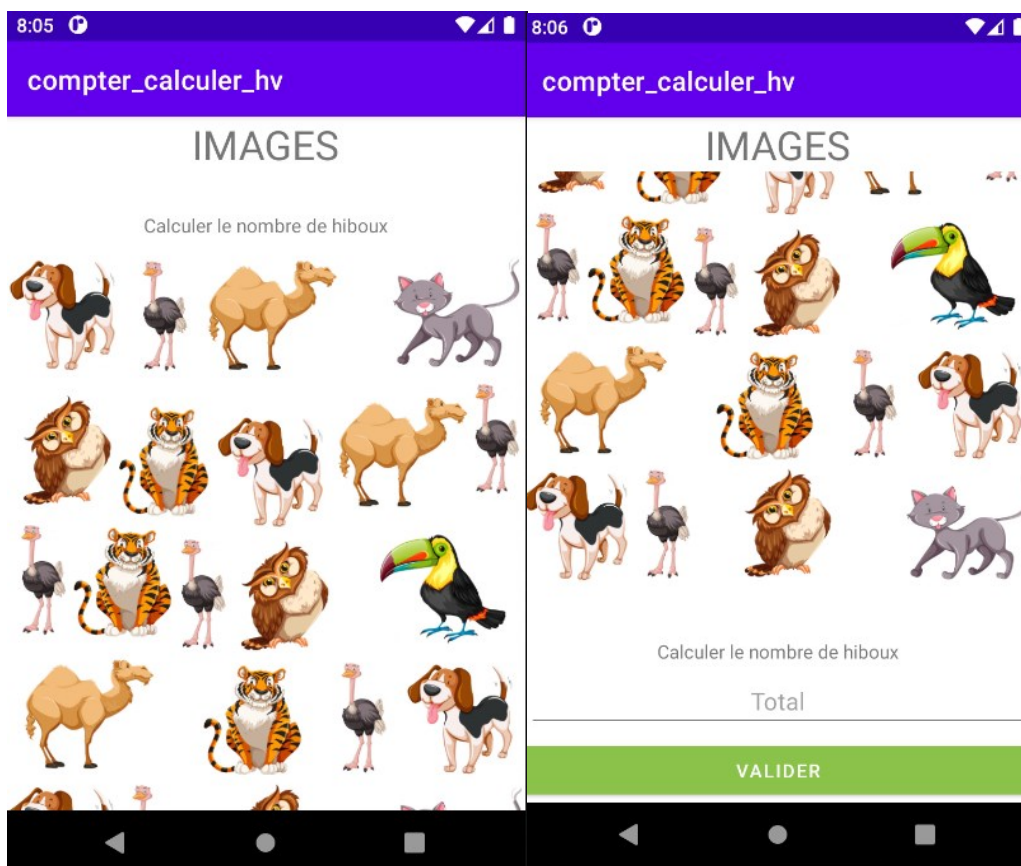
2 usages
String[] animaux_b = {"souris", "chats", "koalas", "girafes", "taureaux", "cerfs", "chevaux"};
1 usage
int[] nombres_b = {5, 4, 6, 3, 2, 3, 4};

2 usages
String[] animaux_c = {"furets", "canards", "crocodiles", "grenouilles", "chenilles", "éléphants", "pandas"};
1 usage
int[] nombres_c = {6, 4, 3, 2, 8, 5, 1};

6 usages
int compteur;
```

```
String consigne = "";
ImageView imageViewImage = (ImageView) findViewById(R.id.ImageViewImage);
Random random = new Random();
int numImage = random.nextInt( bound: 3);
if (numImage == 0) {
    imageViewImage.setImageResource(R.drawable.a);
    int i = random.nextInt(animaux_a.length);
    consigne = "Calculer le nombre de " + animaux_a[i];
    compteur = nombres_a[i];
} else if (numImage == 1) {
    imageViewImage.setImageResource(R.drawable.b);
    int i = random.nextInt(animaux_b.length);
    consigne = "Calculer le nombre de " + animaux_b[i];
    compteur = nombres_b[i];
} else {
    imageViewImage.setImageResource(R.drawable.c);
    int i = random.nextInt(animaux_c.length);
    consigne = "Calculer le nombre de " + animaux_c[i];
    compteur = nombres_c[i];
}
}
```

### Interface obtenue



### Tests et difficultés pour les deux applications

L'application a été testée grâce à l'émulateur intégré dans Android Studio. La problématique rencontrée lors de ces développements a été de faire fonctionner ces applications sur l'ordinateur de base pour les réaliser (bugs dus à des modifications de l'environnement du poste par d'autres utilisateurs du Lycée). J'ai été dans l'obligation de changer de poste de travail, ce qui a eu pour conséquence de prendre beaucoup de retard dans la réalisation de mes deux missions.