Travail pratique 1 : Application android de dessin

Objectifs :

* utiliser les techniques de dessin sur mesure vues en classe
* gérer différents types d'événements
* utiliser des composantes Android tels que des LinearLayout, des ImageView ou des HorizontalScrollView

Ce travail consiste donc à développer une application de dessin simple à l'image de nombreuses applications se trouvant sur Google Play ( je me suis basé sur celle Paint For Android que vous pouvez télécharger sur votre téléphone, de nombreuses autres existent ). En vertu du temps qui nous est imparti, certaines fonctionnalités ont dû être mises de côté mais en le développant vous serez surpris de tout ce qu'on arrive à réaliser. Ce document a pour but de vous guider dans le développement de cette application. En cas de doute concernant les fonctionnalités, référez-vous aux comportements de cette application la majorité du temps. En ce qui concerne son apparence vous avez presque carte blanche, en tenant compte des passages en gras dans le document qui sont obligatoires.

\*\*\* Vous pouvez télécharger l’app sur votre téléphone. Si vous voulez la télécharger sur un émulateur, on doit en créer un qui a Google Play installé

**1. Le modèle de classes**

Comme tout projet orienté objet qui se respecte, **des classes représentant l'état des objets sont nécessaires avant d'attaquer l'interface graphique.**

Des classes telles que : TraceLibre, Oval, Triangle, Forme, Efface, etc.

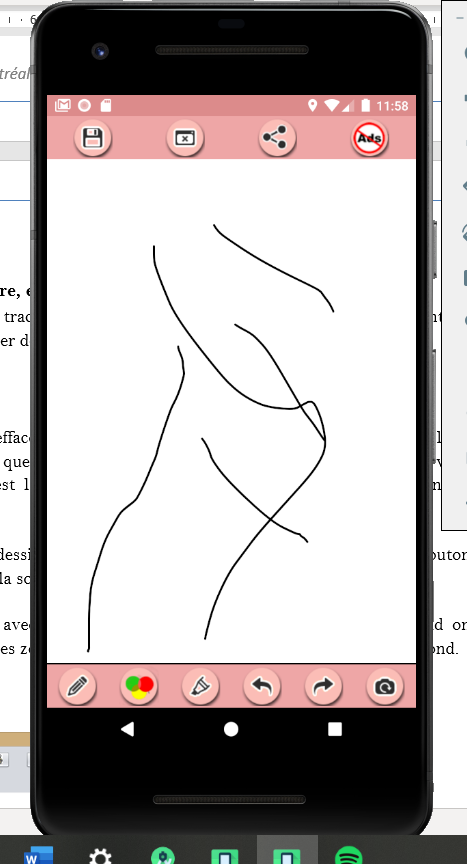
**Pensez à une hiérarchie de classes ( héritage )** . **Pensez également aux éléments / variables d’instance qui permettent de définir les objets de même qu’aux comportements qu’on peut leur associer**.

**2.. L’interface graphique comme tel :**

* **La surface de dessin doit occuper la majeure partie de l’activité.**
* **2 barres d’outils doivent être présentes :**
  + **Une contenant une palette de couleurs à utiliser**
  + **Une contenant les outils qu’on utilise : Crayon, Efface, Rectangle….**
  + **Une des deux doit utiliser un HorizontalScrollView ( ou les deux )**
  + **Un LinearLayout doit être utilisé pour partager l’espace entre la surface de dessin et les barres d’outils, idéalement en utilisant des poids.**

**3. Les outils ( en gras obligatoire, en italique ce sont des bonus )**

* **Outil Crayon** : permet le tracé libre. Pensez à un moyen de relier tous les événements de la souris afin de dessiner des courbes du genre représenté ci-bas ainsi que de les accumuler afin qu'elles soient toutes présentes sur le canvas.



*Plusieurs tracés ont été faits, ils restent sur l’interface…comment ??*

* **Outil Efface** : permet d'effacer les tracés / formes déjà réalisés. **Il s'agit d'un tracé libre mais de la même couleur que la couleur de fond.** Vous devez donc vous assurez que la couleur est la couleur de fond actuelle si vous voulez réellement «  effacer ».

**\*\*\*Attention ! Si on efface avec une couleur identique à la couleur de fond, quand on changera la couleur de fond, les zones effacées changent également pour la couleur de fond afin de demeurer effacées**

* **Outil Pipette** : Permet d'aller chercher la couleur d'un pixel de la surface de dessin pour l'assigner comme couleur pour dessiner soit la couleur utilisée pour le dessin et les formes. Utilisez la méthode suivante afin de créer un objet Bitmap à partir de notre surface de dessin :

**public** Bitmap getBitmapImage() {  
  
 **this**.buildDrawingCache();  
 **bitmapImage** = Bitmap.*createBitmap*(**this**.getDrawingCache());  
 **this**.destroyDrawingCache();  
   
 **return bitmapImage**;  
}

**\*\*\* Notez qu’après l’utilisation de la pipette, l’outil sélectionné doit redevenir le crayon**

* **Outil pot de peinture** : Cet outil permettra de changer la couleur du fond de la surface de dessin en fonction de la couleur sélectionnée. (*Contrairement à l’application de référence Paint for Android, le pot de peinture va seulement changer la couleur de fond, pas la couleur d’une zone fermée, bien que vous pouvez le faire pour un bonus* )

**N'oubliez pas !!! Si on change la couleur de fond., la/les zone(s) effacée(s) doit(vent) changer de couleur aussi**

* **Taille du trait** : un clic sur l’imageView fait apparaître une boîte de dialogue où on peut choisir la taille du trait 🡪 voir Annexe 1 pour TP1
* *Outil enregistrer* ( BONUS ) : À l'aide de la méthode écrite précédemment, on peut avoir l'image correspondante à notre dessin et la sauvegarder sur le téléphone…
* **Rectangle** : pour dessiner un rectangle vide se redimensionnant au fur et à mesure
* **Cercle** : pour dessiner un cercle vide se redimensionnant au fur et à mesure
* **Triangle** : pour dessiner un triangle vide se redimensionnant au fur et à mesure. Ce dernier item sera plus difficile à réaliser. On commence par dessiner un segment entre les deux premiers sommets, on enlève notre doigt et on clique à nouveau pour créer le 3ème sommet.

**4. Barre d’outils couleur**

Utiliser au moins 8 couleurs, je les ai affichées dans des TextViews ou des Buttons. Trouvez un moyen d’afficher en tout temps la couleur en vigueur.

*"nuage de couleur":* ( BONUS )Outre les couleurs principales ci-haut, utilisez une librairie en ligne permettant de choisir n'importe quelle couleur. Cette roue/nuage de couleurs/*colorPicker* apparaîtra dans une boîte de dialogue à partir de l'activité principale.

**3. Conseils**

* De nombreux événements et de nombreuses sources sont présents. Soyez rigoureux dans votre approche afin de ne pas vous perdre. Restreignez votre nombre d'écouteurs ( personnellement, j'en ai 2 )
* Le principal travail est d'accumuler les anciens tracés/formes afin de toujours les redessiner, les garder présents dans la surface de dessin.
* Commentez bien votre code, donnez des noms significatifs à vos variables / composantes
* Attention à ne pas répéter les mêmes opérations ( ajout de sources à un écouteur, vérifier quelle couleur a été sélectionnée )
* Assurez-vous que votre application se comporte bien. Dans le doute, référez-vous aux comportements de l’app de référence la plupart du temps ou posez-moi des questions.

**Évaluation :**

|  |  |
| --- | --- |
| Formative obligatoire | 1er octobre |
| Sommative | 8 octobre |
|  |  |

**Grille d'évaluation**

|  |  |  |
| --- | --- | --- |
| Modèle de classes | 15% |  |
| Apparence du GUI | 10% |  |
| Fonctionnalités de dessin | 70% |  |
|  |  |  |
| Commentaires | 5% |  |
| Bonus possibles ( autres fonctionnalités de l’application, annuler Undo / Redo , formes pleines, enregistrer, chaque élément est dessiné avec une couleur aléatoire etc. ) | Maximum de + 10% |  |

**Plagiat**

Le logiciel MOSS développé à l’université Stanford sera utilisé <https://theory.stanford.edu/~aiken/moss/>. Si la comparaison de deux travaux identifie un nombre de lignes identiques considérablement supérieur à la moyenne des comparaisons des autres paires de travaux, les deux travaux obtiendront la note 0.

**Commentaires / recours à l’IA**

Votre code doit être commenté de manière à comprendre votre démarche, surtout dans les algorithmes. Si je ne comprends pas un passage de votre code ( car il n’est pas commenté en autres ) en relation avec la matière vue au cours, j’enlèverai ce code de votre application et les points s’y rattachant.