Travail pratique 2 : Annexe 1

Éléments pour se préparer au TP :

* Classe Path : pour réaliser des traits / courbes
* Classe AlertDialog : pour faire des fenêtres de dialogue

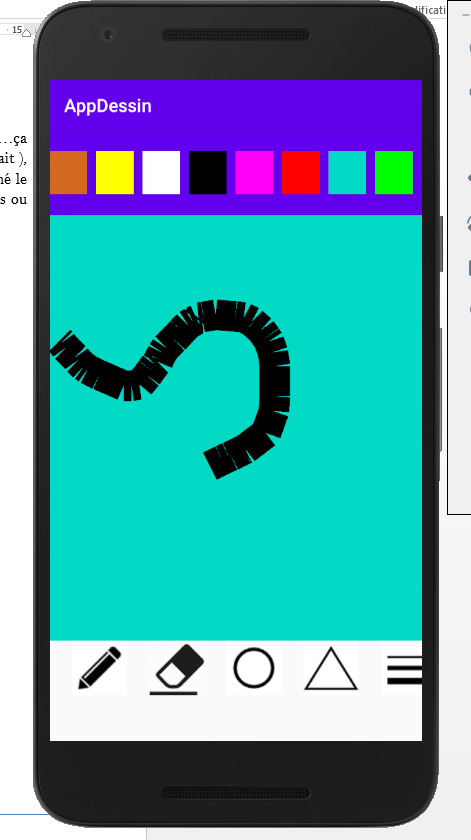
Dans les classe il faut savoir plusieurs chose par exe,ple la couleur peut être dans une super class puisqu`elle et une variable commune, la largeur du traint etc faire les modeles de class en premier

Pour effacer dessiner uen ligne avec la couelur de fond

Dans les class ont peut mettre une classe dessiner(canvas c) et caller les fonction dans l’activity

## Classe Path

On pourrait décider de considérer un Vector de Points comme variable d’instance de certaines classes-modèle. Ensuite, on pourrait relier les points recueillis à l’aide la méthode drawLine…ça fonctionne jusqu’à un certain point. En effet, si on élargit le segment ( la taille du trait ), le rectangle dans lequel le segment est dessiné est également agrandi et, étant donné le nombre élevé de points et leur changement de direction, on obtient un résultat plus ou moins satisfaisant :



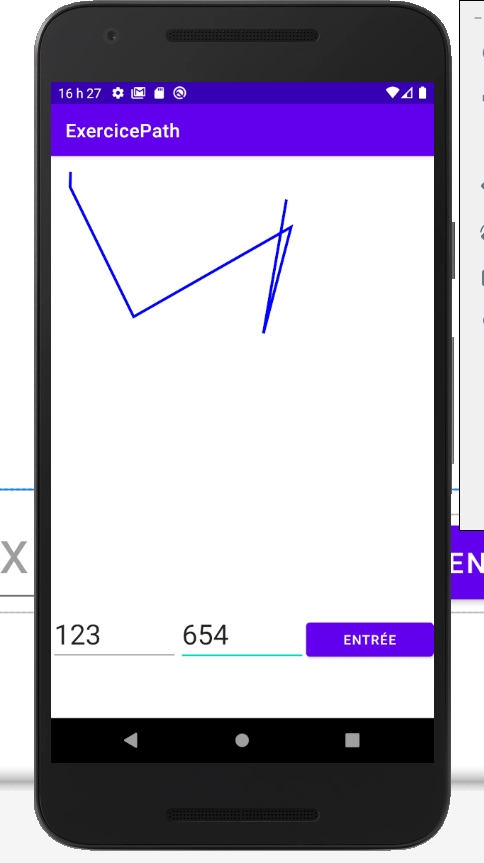
Solution : Utiliser un objet Path

* Créer un objet Path
* Début du tracé : méthode moveTo
* Relier des points : méthode lineTo
* Dessiner l’objet : canvas.drawPath ( objet Path, objet Paint )



## Exercice:

Dans un nouveau projet ayant un LinearLayout comme conteneur principal, ajoutez à la fois une surface de dessin occupant les quatre cinquièmes de la hauteur de l’activité ainsi qu’une zone regroupant des contrôles :



Faites la gestion sur le bouton « entrée » de manière à ce qu’un objet Path, créé initialement, se compose peu à peu à partir des points dont les coordonnées sont entrées dans les champs texte

## Classe AlertDialog

On veut pouvoir faire en sorte qu’une boîte de dialogue apparaisse lorsqu’on clique sur le bouton/ImageView pour les largeurs de trait :



Et idéalement qu’on puisse récupérer cette valeur dans notre activité. Le processus est légèrement complexe mais heureusement AndroidStudio peut nous aider ( et moi aussi ! ) Voici une marche à suivre. seekbar

1. On crée une Activity comme à l’habitude qui représentera notre boîte de dialogue. On peut le créer en faisant sur le nom du package ( New 🡪 Activity 🡪 Empty Activity )

Renommez la superclasse Dialog plutôt que AppCompatActivity. On vous demandera alors de générer un constructeur ce que vous devez faire avec la suggestion.

1. Dessinez le fichier de positionnement xml représentant l’apparence de votre boîte de dialogue :

Assurez-vous que votre layout conteneur aie des constantes « wrap\_content » comme dimension et non pas « match\_parent »

Une image contenant texte, ordinateur, capture d’écran, intérieur

Description générée automatiquement

1. Codez le onCreate habituel, donc vous pouvez y initialiser vos composantes, initialiser votre écouteur.

Dans cette classe, vous pouvez faire la gestion d’événement propre à votre seekBar et au bouton.

1. Finalement, de retour dans votre Activité, on instancie la boîte de dialogue et on l’affiche avec la méthode show. Pensez comment vous pouvez référer à l’activité parent à partir de votre boîte de dialogue afin de passer la valeur choisie comme largeur de trait