

Rapport de développement

Sujet : Hyperstack

Préambule	2
Architecture générale du projet	3
Le fonctionnement des options de dessin	4
Organisation adoptée pour le développement	5
Difficultés rencontrées	6
Solutions adoptées	7

Préambule

Le groupe est constitué de 6 membres :

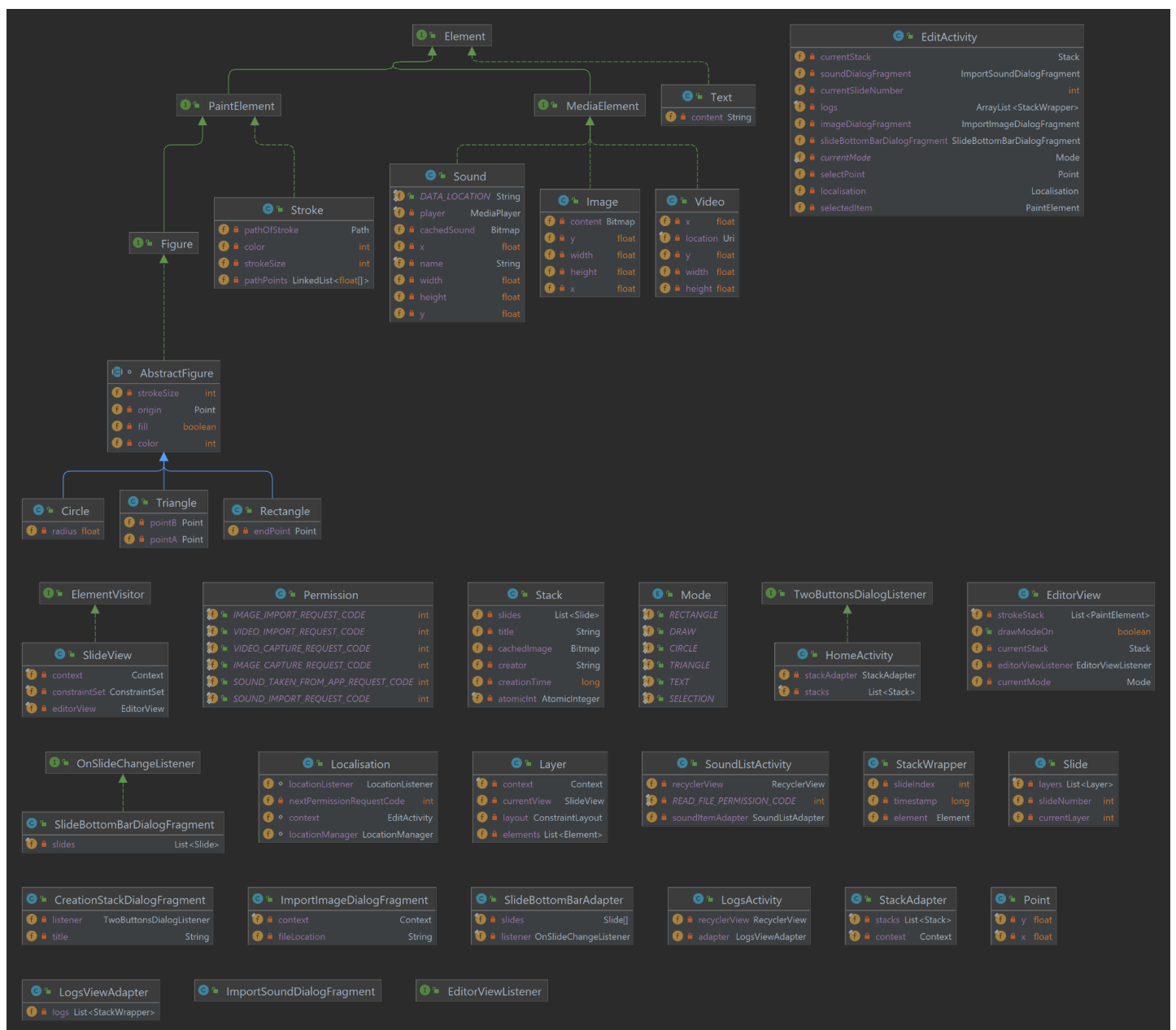
- Anthony LIM
- Florian CAHAY
- Kévin SAVANE
- Kosal LUC
- Michel IP
- Vincent LIN

Le sujet choisi est Hyperstack qui consiste à réaliser une application gérant une pile de planches interactives avec des liens hypertextes.

Architecture générale du projet

Tous les éléments que l'on peut ajouter à une diapositive doivent implémenter l'interface **Element** où une de ses sous interface (**PaintElement**, **MediaElement**). Les éléments qui implémentent l'interface **PaintElement** sont des éléments qui peuvent être dessinés dans un **Canvas**. Alors que ceux qui implémentent **MediaElement** sont ceux qui doivent être ajoutés via une **View**.

Une présentation (**Stack**) contient une liste de diapositives (**Slide**). Chaque diapositive contient à son tour une liste de calques (**Layer**). Et enfin chaque calque contient une liste d'éléments.



Le fonctionnement des options de dessin

Il y a 3 catégories d'élément que l'on peut dessiner sur l'application :

- Les figures géométriques comme le rectangle, le triangle ou le cercle

Pour dessiner une figure il faut d'abord passer en mode dessin, c'est-à-dire utiliser la méthode `setFigureSetup()` pour définir le Listener de l'activité.

Quand le listener reçoit une entrée, il va initialiser la figure à dessiner, puis suivant le déplacement de l'appui de l'utilisateur la figure va s'agrandir ou se rétrécir. On observe une prévisualisation de la figure car on met à jour la View à chaque déplacement.

Toutes les figures sont dessinées sur la même View.

- Dessin à main levée

Il est possible de dessiner des croquis avec un mode "Crayon", le `onTouchListener` de la View est configuré de la même manière que pour les figures géométriques. On observe également une prévisualisation du tracé. Toutes les figures sont dessinées sur la même View.

- Le contenu multimédia comme les images, son ou vidéo.

Pour ajouter du contenu multimédia, il suffit d'accéder au menu qui est présent sur l'activité d'édition.

Chaque élément correspond à une View.

La possibilité de sélectionner une figure et de la déplacer a aussi été mise en place. Il faut tout d'abord passer en mode sélection puis effectuer un appui sur l'écran, une fois cela effectué le code va aller chercher sur la stack si le point appartient à l'une des figures présentes. Si une figure a été sélectionnée, celle-ci pourra être déplacée, la distance de déplacement de la figure correspond à l'écart entre le point de sélection et le point de relâchement.

Organisation adoptée pour le développement

La charge de travail du projet étant conséquente, nous avons choisi de mettre en place un [Kanban](#) en ligne sur [Asana](#) pour nous aider à nous répartir les tâches. La première étape a été de découper le projet en plusieurs tâches réalisables. Étape qui est un peu compliquée au début du projet car il faut bien comprendre le besoin et réussir à le découper en catégorie.

Suite à la découpe du projet en tâches, nous nous les sommes attribués en fonction de ce qu'on avait envie de faire.

L'attribution des tâches est la suivante :

- Anthony :
 - Ajout du dessin à main levée
- Florian :
 - Importation d'une image
 - Importation d'une vidéo
 - Écran d'accueil
 - Interface écran d'édition
- Kévin :
 - Ajout des éléments textes
 - Suppression des éléments d'une slide
 - Suppression d'une slide
 - Navigation entre les slides
 - Ajout d'une slide
- Kosal :
 - Importation d'un son
- Michel :
 - Enregistrement des actions de l'utilisateur
 - Sérialisation des Piles
- Vincent :
 - Ajout de figure géométrique
 - Ajout de la sélection et du déplacement des figures

Le suivi du développement de ce projet s'est fait via GitHub

(<https://github.com/FlorianCahay/HyperStack>) où on retrouve les commit de chacun.

On échangeait aussi via discord pour poser toutes sortes de questions aux autres membres du groupe.

Parfois, nous avons travaillé en [pair programming](#) ce qui nous permettait de nous entraider sur certains points compliqués.

Difficultés rencontrées

- La charge de travail du projet
 - Le projet est conséquent, ça a été difficile de découper le projet en différentes catégories afin de nous répartir le travail.
- La gestion des Intent
 - Pour pouvoir se déplacer d'une activité à une autre, il faut s'assurer que :
 - Les données qui vont être transférées d'une activité à une autre doivent être sérialisables
 - Par suite, il faut alors que les champs le soient aussi
- Le manque de motivation
- Changement de la structure du projet
 - Nous avons décidé de changer la structure du projet au niveau de l'affichage des éléments sur une slide au milieu du projet. On s'est rendu compte que l'ancienne méthode ne fonctionnait qu'avec les croquis et les images. Cela ne suffisait pas pour les autres types d'éléments.
- La communication entre Activity, Fragment et Adapter
 - Dans certaines situations, nous avons eu besoin d'envoyer des données depuis un Adapter vers une Activity, en passant par un Fragment. Etant donné que nous ne maîtrisons pas la programmation android à la perfection, ce fut un problème relativement difficile à résoudre.
- La sauvegarde des données

Solutions adoptées

- Mise en place d'un Kanban en ligne pour la répartition des tâches.
- Utilisation du pattern Visitor pour gérer l'affichage des éléments sur les diapositives.
- Utilisation d'Interface et de Listener pour transmettre les informations entre Activity, Fragment et Adapter.
- Construction d'une classe Wrapper pour garantir un objet sérialisable et qui contient des objets sérialisables, afin de pouvoir les ajouter dans les Intent.