Cahier d’analyse

Préparé pour  : [Sebastien Lefevre](http://foad.univ-ubs.fr/user/view.php?id=30911&course=1)

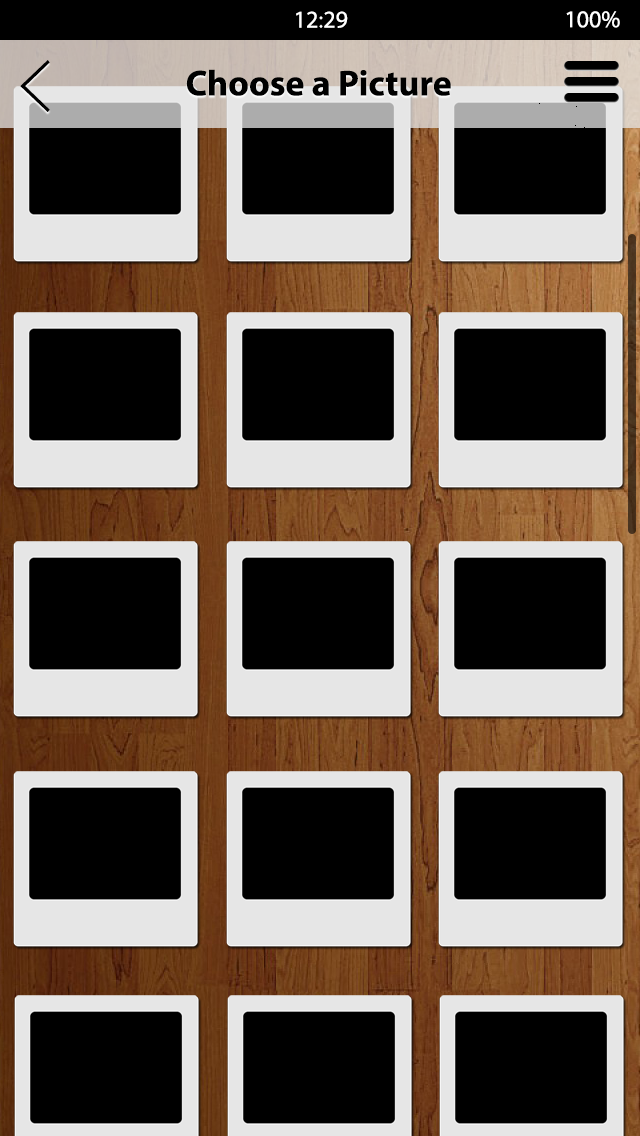
Préparé par  : Rouillé Jérôme, Loriot Sacha, Haddad Mehdi, Catiau-Tristant Florent

Photo boost

DATE \@ "d MMMM y" 22 avril 2014

Thème du projet : Logiciel de retouche photo pour PDA

Sommaire

photo boost

TOC \t "Titre, 1,Titre 2, 2"

1. Objectif du projet PAGEREF \_Toc \h 3
2. Description des fonctionnalités PAGEREF \_Toc1 \h 4
3. L’avancement PAGEREF \_Toc2 \h 6
4. Diagramme de séquence boite noire PAGEREF \_Toc3 \h 7
5. Diagramme de classe d’analyse PAGEREF \_Toc4 \h 9
6. Scénarios de test PAGEREF \_Toc5 \h 11
7. Planning prévisionnel PAGEREF \_Toc6 \h 13
8. Objectif du projet

Ce projet a pour but la réalisation d’une application Java avec l’aide du PDA. Grâce à nos connaissances nous avons réalisé un diagramme d’analyse qui nous serviras de modèle de conception du logiciel, des diagrammes de séquences boite noire pour visualiser l’interaction avec l’utilisateur et une interface Homme-Machine pour avoir un visuel du logiciel. Bien évidemment nous avons préalablement réfléchi sur les fonctionnalités du logiciel.

photo boost

Suite à ça, nous devons entamer la phase de codage du logiciel qui va prendre la majeur partie du temps. En effet, nous allons devoir tester, débuguer et optimiser notre code afin de faire fonctionner notre logiciel au mieux.

Lors de la réalisation de ce projet, nos connaissances d’expressions sont mis en jeu. Nous devons réaliser un cahier d’analyse, une présentation orale et mettre en oeuvre un travail de groupe organisé et efficace.

1. Description des fonctionnalités

**PARTIE RÉGLAGES**

• Ecraser ou nouvelle image par défaut

• Méthode : equals() (test d’égalité d’images)

• Type de galerie :

o Aperçue d’image sous forme de carrés

* 2 ou 3 colonnes

o *Coverflow (OPTIONNEL)*

**PARTIE RETOUCHE**

Headbar:

* + - * **Retour :** Annule la dernière action appliquée.
      * **Rétablir :** Rétablie une action annulée.

photo boost

* + - * **Appliquer :** Sauvegarde l’image (ouvre une pop-up “Ecraser ou créer nouveau fichier ?”).
      * **Annuler :** Annule toutes les modifications de l’image jusqu’à la précédente sauvegarde.
      * **Quitter :** Reviens à l’écran d’accueil (pop-up “voulez-vous enregistrer ?”).

Tabbar:

Effets :

* + - * **Flou (Netteté globale) :** Diminue la netteté de l’image.
      * **Pixélisation :** Les zones de couleur homogène passe au format d’un pixel. Flou pixelisé.
      * **Augmenter netteté, contrastes :** Accroit les nuances, diminue le flou.
      * *(OPTIONNEL)* ***Opacité :*** *Augmente la transparence de l’image.*
      * *(OPTIONNEL)* ***Filtre couleur :*** *Applique une augmentation de la couleur choisie (ex : image bleutée).*
      * *(OPTIONNEL)* ***Cadre :*** *Un cadre sera ajouté à l’image sélectionnée. Cependant, ce cadre ne devra en aucun cas faire obstruction à quelconque partie de l’image choisie. Nous définirons des cadres. (Nous devons nous occuper de l’adaptation du cadre à une image.)*

Sélection :

* + - * **Lasso :** Sélectionne selon une forme non régulière (suis le tracé du pointeur). Possibilité de choisir une sélection interne ou externe à la zone formée.

photo boost

* + - * *(OPTIONNEL)* ***Baguette magique :*** *Sélection automatique de pixels selon les zones de couleur homogènes.*

Filtres :

* + - * **Sépia :** Colore l'image avec des nuances de bruns, donne un effet vieillis.
      * **Black & White :** Enlève les couleurs de l’image, devient noire & blanche.
      * **Bande dessinée :** Applique un filtre de style bande dessinée à l’image.
      * **Négatif :** Inverse les couleurs d’origine de l’image (par exemple le rouge devient cyan).

Sizing :

* + - * **Redimensionner :** Rognage manuel (barres verticales et horizontales qui délimitent le rognage) ou Rognage automatique (4/3 ou 16/9).
      * **Rotation :** Permet de faire une rotation de l’image à 90°,180° ou 270°.
      * *(OPTIONNEL)* ***Miroir :*** *Applique une rotation sur un axe vertical pour afficher le reflet de l’image.*

1. L’avancement

**D**é**j**à **r**é**alis**é **:**

• Diagrammes de séquence boite noire

• Représentation de l’interface graphique

• Enumération des fonctionnalités de l’application

• Diagramme de classe d’analyse

**A r**é**aliser :**

• Diagramme de classe fonctionnel

• Codage

**Outils utilis**é**s lors de la phase de conception en amont :**

• Visual Paradigm for UML (Diagrammes)

• Photoshop (Réalisation de l’interface)

photo boost

• Google drive (mise en commun des documents)

**Outils pr**é**vus pour la suite du projet :**

• Langage de programmation : Java

• Eclipse ?

**Connaissances d**é**velopp**é**es depuis le d**é**but du projet :**

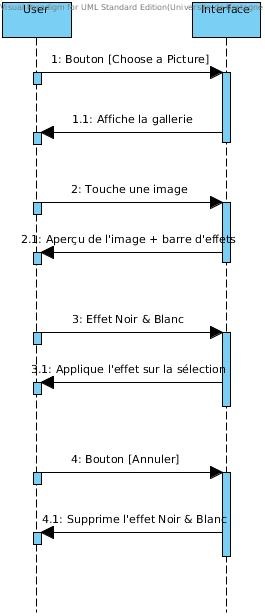
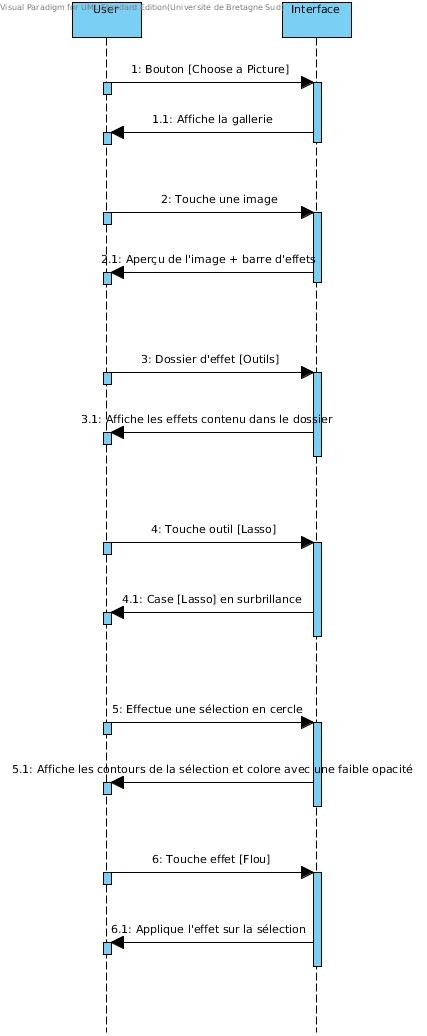
• Gestion de groupe, organisation des tâches

• Respect d’un projet

• Réflexion sur un diagramme de classes d’analyse à partir de peu

1. Diagramme de séquence boite noire

photo boost



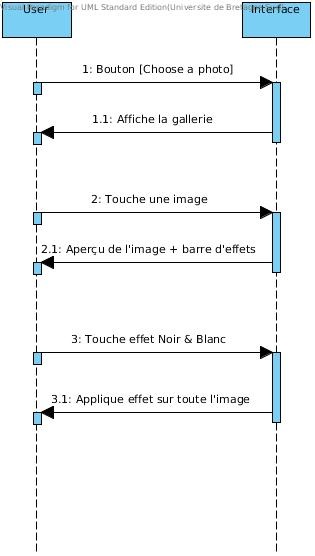
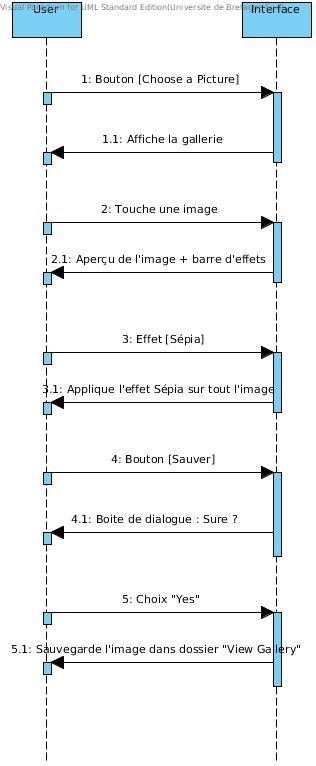


photo boost

1. Diagramme de classe d’analyse

photo boost

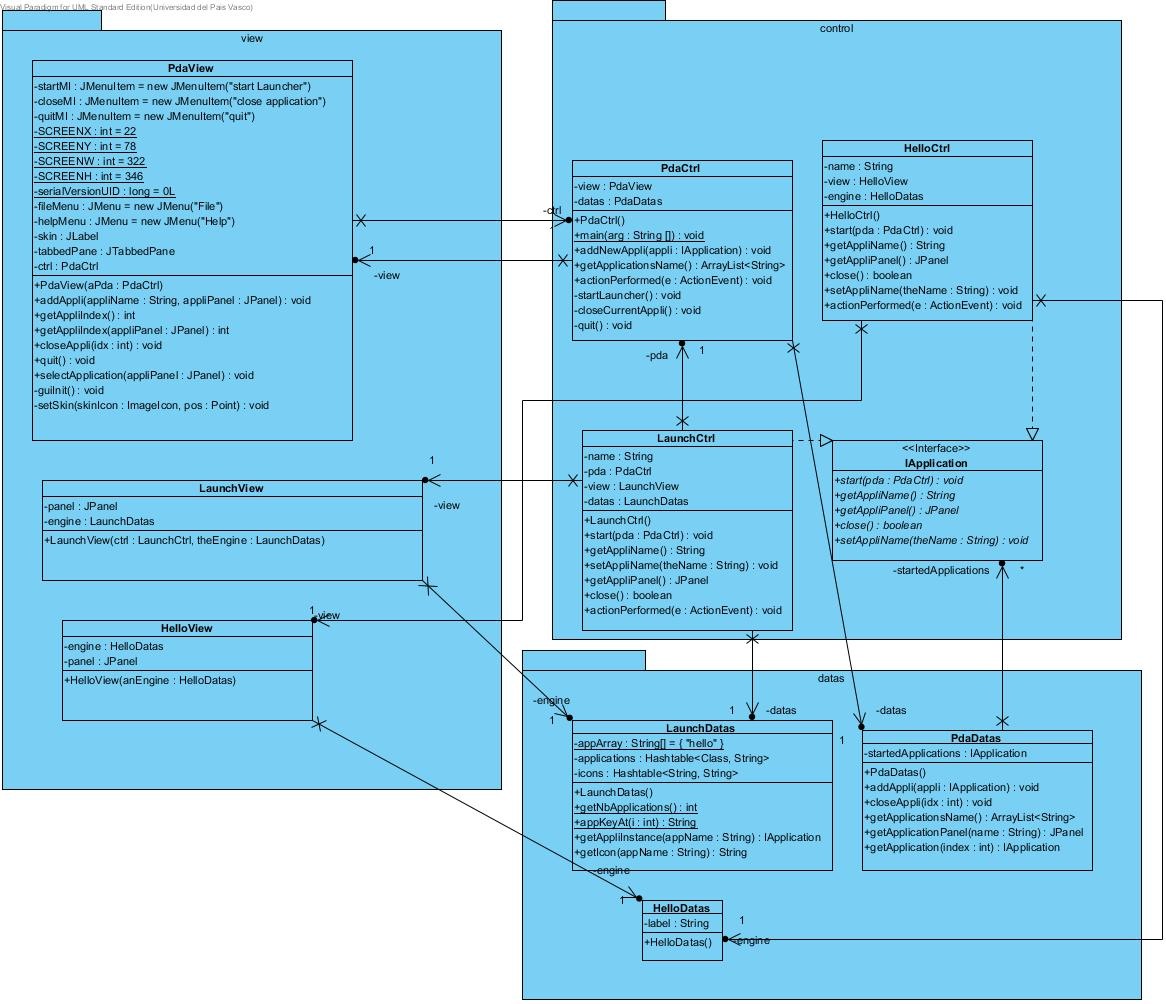
Retro-conception du PDA :

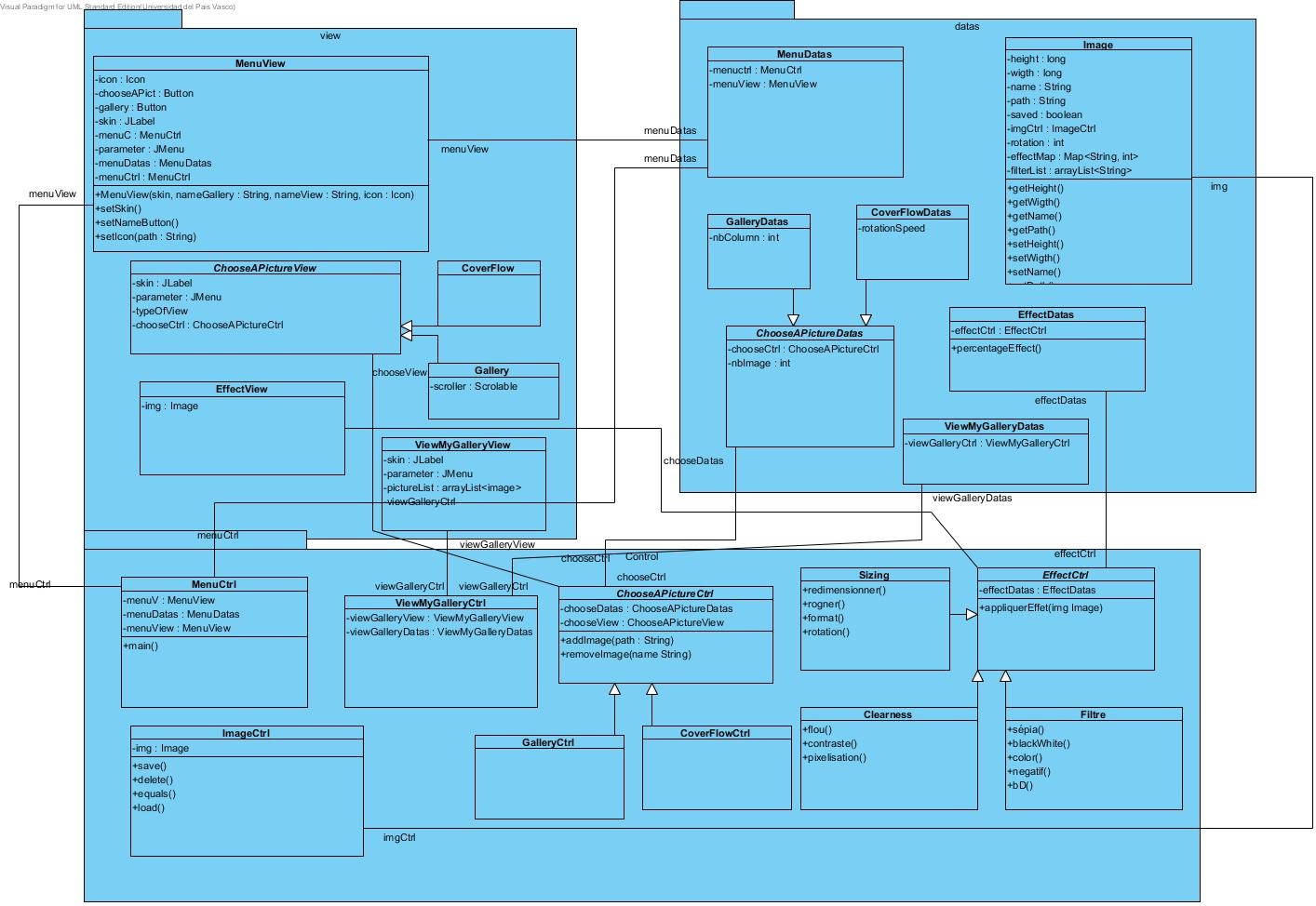
Diagramme de classe du logiciel :

photo boost

1. Scénarios de test

I) Annuler une action

Pré-requis :

Image chargée

Effet flou (50%) appliqué

Test :

Clic sur bouton annuler dernière action

Résultat :

Être précis sur le résultat attendu : Comment vérifier le résultat ?

Résultat méthode equals() entre l’image d’origine et la nouvelle = true

II) Rotation 90°

Pré-requis :

Image chargée

Test :

Effet rotation (2 fois)

photo boost

Résultat :

attribut ‘rotation’ = 180°

résultat equals() entre l’image d’origine et la nouvelle = false

III) Paramètre 3 colonnes sur la gallerie

Pré-requis :

Etre sur la page ‘gallery’ avec paramètre 2 colonnes ou Coverflow

Test :

Cliquer sur paramètre (icone en haut à droite)

Choisir sur la partie ‘gallery’ : 3 colonnes

Résultat :

3 colonnes d’affichées

attribut nbColumn de la class GalleryDatas = 3

attribut typeOfView de la class ChooseAPictureView = “gallery”

IV) Appliquer un filtre

Pré-requis :

Image chargée.

Test :

Appliquer un filtre quelconque. Prenons ici pour exemple le filtre “Sépia”.

Résultat :

L’image est désormais modifiée avec un filtre. Pour le vérifier, regardons de plus près l’attribut “filterList” de la classe “Image”. Celui-ci doit contenir la chaîne de caractère “Sepia”. De plus nous devons exécuter la méthode “equals” afin de s’assurer que l’image précédente et l’image avec le filtre sepia sont belle et bien différente. Equals() doit donc renvoyer “false”

photo boost

V) Appliquer un effet

Pré-requis :

Image chargée.

Test :

Appliquer un effet quelconque. Prenons ici pour exemple l’effet “Flou”.

Résultat :

L’image est désormais modifiée avec un filtre. Pour le vérifier, regardons de plus près l’attribut “effectMap” de la classe “Image”. Celui-ci doit contenir la chaîne de caractère “Flou” ainsi que le pourcentage auquel nous avons poussé l’effet lorsque nous avons appliqué ce dernier. De plus nous devons exécuter la méthode “equals” afin de s’assurer que l’image précédente et l’image avec l’effet “Flou” sont belle bien différente. Equals() doit donc renvoyer “false”.

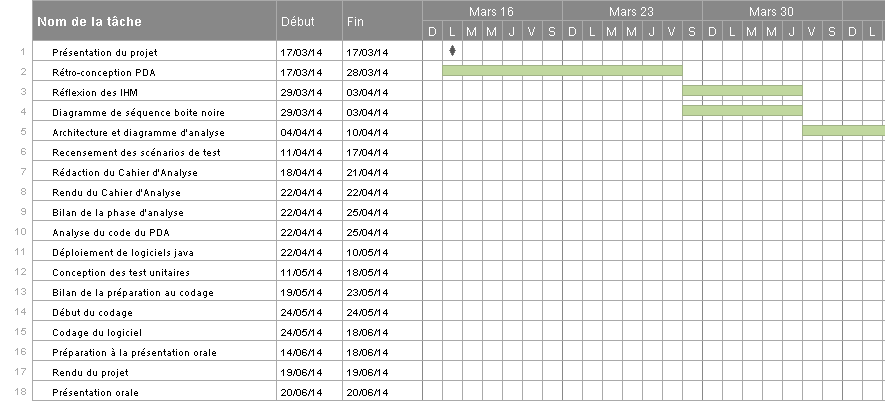
1. Planning prévisionnel

photo boost

