|  |
| --- |
| Helias Maxime  Guerry David |
| Logiciel de Dessin |
| Projet Java |

|  |
| --- |
| 01/04/2015 |

Table des matières

**Présentation**

Equipe 2

Présentation du projet 2

**Le Projet**

Analyse3

Diagramme de classe 3

Architecture du programme 3

Répartition des tâches 4

David 4

Maxime 10

Présentation

# Equipe :

Notre équipe de projet est composée de deux étudiants. Maxime Helias, 21 ans, et David Guerry, 22 ans, tous les deux en formation à la faculté de sciences de Poitiers.

Ce projet nous a été assigné en cours de Programmation Orientée Objet avec Peltier Samuel et Sylvie Alayrangues.

# Présentation du projet :

Le but de ce projet est de réaliser un petit logiciel de dessin.

Il existe 2 types d'images :

- Les images couleurs, pour lesquelles, chaque forme est associée à 2 couleurs, celle du remplissage et celle du contour.

- Les images ASCII, pour lesquelles, chaque forme est associée à 2 caractères, celui du remplissage et celui du contour.

Pour les formes ASCII, chaque caractère correspond à un pixel.

Pour chaque forme couleur (resp. ASCII), il est possible d'inverser la couleur (resp. le caractère) de remplissage et la couleur (resp. le caractère) de bordure.

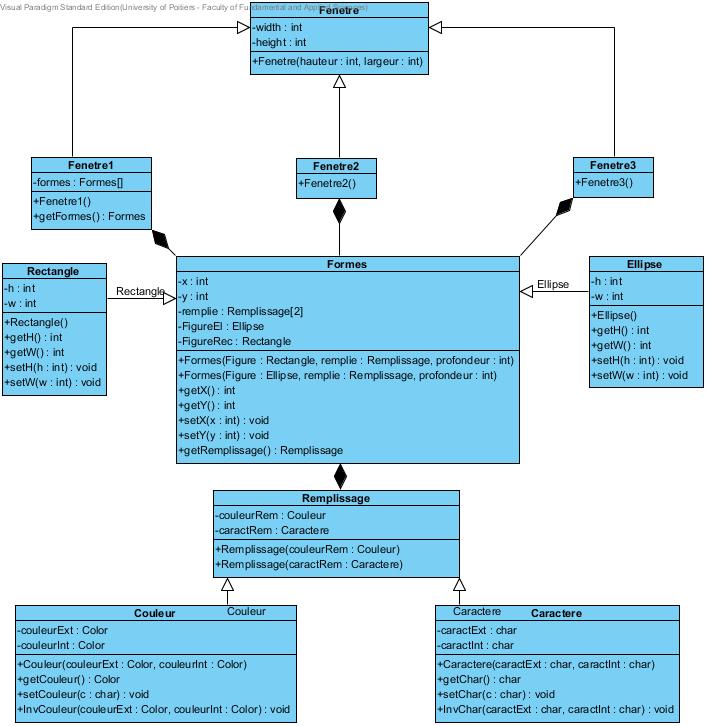
Une forme peut être dessinée en version remplie ou en version fil de fer (dans ce dernier cas, seul son bord est dessiné).

Chaque forme est également positionnée a une certaine profondeur dans une image. Une forme à la profondeur 0 est au premier plan. Une forme à la profondeur 1 est juste derrière et ainsi de suite.

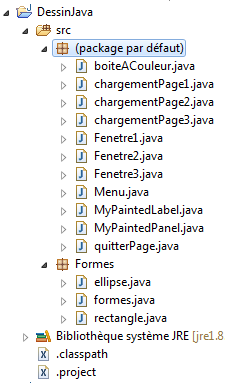
Projet

# Analyse

Diagramme de class



Architecture du programme

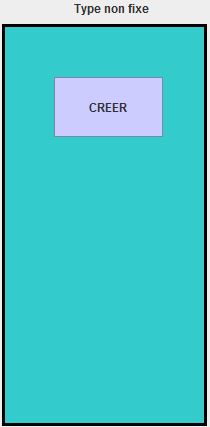


# Répartition des tâches

David

David c’est occupé de la partie développement des interfaces utilisateur. Pour se faire, il a dû commencer par créer les mock-up de l’application en commençant par l’interface d’accueil du programme. Notre programme comporte 4 interfaces différentes. La première correspond à l’interface d’accueil du programme (La première fenêtre qui se lance leur de l’exécution du programme.) Sur cette fenêtre, nommé  « Menu », l’utilisateur aura la possibilité de choisir dans quel mode il veut lancer le programme. En effet, le programme propose 3 modes de dessin différents qui sont :

* Le mode Forme fixe
* Le mode Forme semi-fixe
* Le mode Forme non fixe



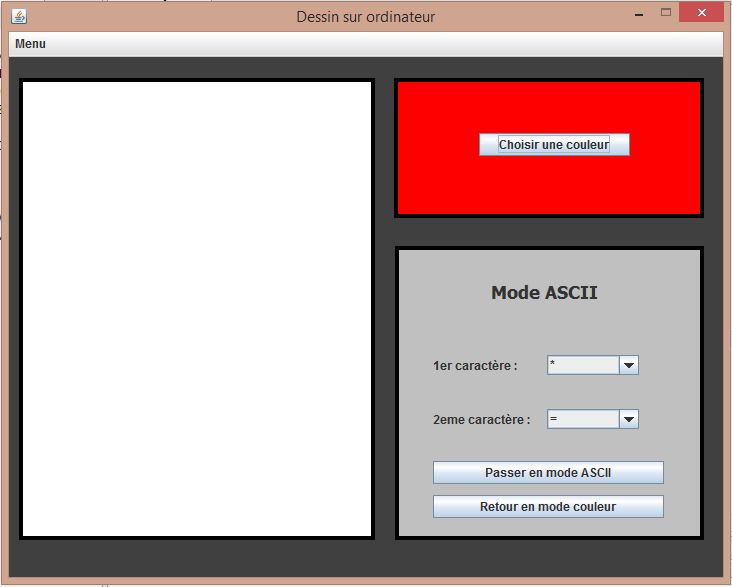
Pour le mode forme fixe, l’utilisateur devra choisir entre l’option fil de fer (ce qui lui dessinera juste les contours de la forme) ou l’option remplie (ce qui lui dessinera une forme remplie). De plus, il devra choisir la forme qu’il veut dessiner. Les formes possibles seront de deux :

* La forme rectangle
* La forme ellipse

Par défaut, c’est la forme rectangle qui est sélectionné.

Une fois les options choisies, l’utilisateur pourra valider en appuyant sur le bouton « CREER » ce qui lui ouvrira une nouvelle fenêtre. Dans cette fenêtre, l’utilisateur pourra commencer à dessiner mais seulement avec les options précédemment choisie. Il ne pourra, en aucun cas, modifier son choix de remplissage ainsi que son choix de forme.

*Interface du mode forme fixe :*



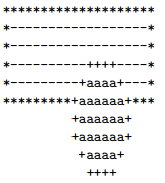
Cette interface est composée de 3 parties distinctes.

La première, à gauche et en blanc, représente la partie où l’utilisateur pourra dessiner.

La deuxième, en haut à droite et en rouge, représente la partie où l’utilisateur pourra choisir la couleur de ces formes. Lors de la sélection de la couleur, la couleur de font du panel se modifiera en la couleur sélectionné et donc, celle de la futur forme. Par défaut, la couleur est rouge.

La troisième, en bas à droite et en grise, représente la partie où l’utilisateur pourra passer en mode ASCII. Ce mode ascii permet de remplacer la couleur par des formes.

Ex :



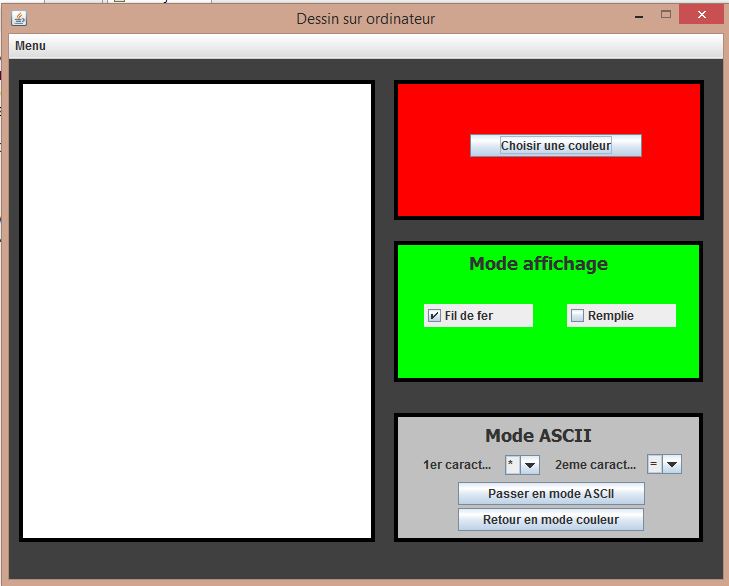
Pour le mode forme semi-fixe, l’utilisateur devra choisir la forme qu’il veut dessiner. Les formes possibles seront toujours les mêmes :

* La forme rectangle
* La forme ellipse

Par défaut, c’est toujours la forme rectangle qui est sélectionné.

Une fois l’option de la forme choisie, l’utilisateur pourra valider en appuyant sur le bouton « CREER » ce qui lui ouvrira une nouvelle fenêtre. Dans cette fenêtre, l’utilisateur pourra commencer à dessiner. Encore une fois, il ne pourra, en aucun cas, son choix de forme.

*Interface du mode forme semi-fixe :*



Cette interface est composée de 4 parties distinctes.

Nous retrouvons les 3 parties de la précédente interface qui sont :

La première, à gauche et en blanc, représente la partie où l’utilisateur pourra dessiner.

La deuxième, en haut à droite et en rouge, représente la partie où l’utilisateur pourra choisir la couleur de ces formes. Lors de la sélection de la couleur, la couleur de font du panel se modifiera en la couleur sélectionné et donc, celle de la futur forme. Par défaut, la couleur est rouge.

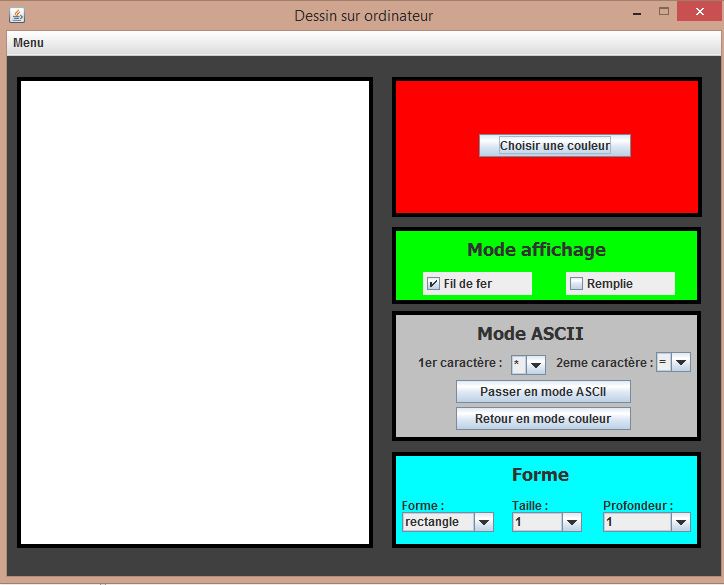
La troisième, au milieu à droite, représente la partie où l’utilisateur pourra choisir l’option d’affichage de ses formes. Il a deux possibilités :

* L’option remplie
* L’option fil de fer

La quatrième, en bas à droite et en grise, représente la partie où l’utilisateur pourra passer en mode ASCII. Ce mode ascii permet de remplacer la couleur par des formes.

Pour le mode forme non fixe, l’utilisateur n’aura aucune option à choisir, toutes les différentes options sont intégré dans la fenêtre qui s’ouvrir lors de la validation en appuyant sur le bouton « CREER ».

*Interface du mode forme non fixe :*

**

Cette interface regroupe les différentes options des deux autres. Le seul changement est la possibilité de modifier la forme du dessin, la taille et la profondeur.

Maxime

Maxime c’est occupé de la partie dessin, mais avant de se lancer directement dans le développement de l’application, il a commencé par faire des programmes de test.

Le premier était sur l’écoute de la souris, afin de récupère ces cordonnes en utilisant les différentes méthodes existant (mousePressed, mouseDragged, mouseReleased, mouseClicked, etc…).

La deuxième étape fut de construire un rectangle avec les composants Graphics. La technique utilisé a était de surcharger la méthode : paintComponent(Graphics g), et d’utiliser les méthodes de la classe MouseAdapter, vue à la première application de test, afin d’y extraire les coordonnées de la souris et de dessiner notre rectangle grâce à ces points.

Après cela, nous avons pu mettre en application ces différents tests dans notre programme en les intégrant dans une classe MyPaintedPanel. Cette classe créera notre panel de dessin. Afin de pouvoir dessiner plusieurs formes, il a fallu améliorer la classe de façon a lister chaque forme. On a choisi de créer une classe MyPaintedLabel, qui contiendra toutes les informations nécessaire d’une forme, et qui créera un label pour chaque forme. Chaque formes ou objet MyPaintedLabel sera stocké dans un « ArrayList » puis redessiné à chaque ajout d’une nouvelle forme en parcourant l’ArrayList. Il faudra appeler la méthode qui dessine nos formes.

La classe MyPaintedLabel stock pour chaque objet les informations suivantes :

**private** Point origin;

**private** Point end;

**private** Color couleur;

**private** String typeForme;

**private** **boolean** rempli;

**private** **int** contEpaisseur;

Selon le type de forme à dessiner et si elle est en mode fil de fer ou rempli, on appelle un objet Graphics différents

**if**(rempli == **false**){

Graphics2D g2 = (Graphics2D) arg0;

Ellipse2D shape1 = **new** Ellipse2D.Float(x1, y1, x2-x1, y2-y1);

Stroke s = **new** BasicStroke(contEpaisseur+.0f, // Width

BasicStroke.*CAP\_SQUARE*, // End cap

BasicStroke.*JOIN\_MITER*, // Join style

10.0f, // Miter limit

**null**, // Dash pattern

0.0f); // Dash phase

g2.setStroke(s);

g2.draw(shape1);

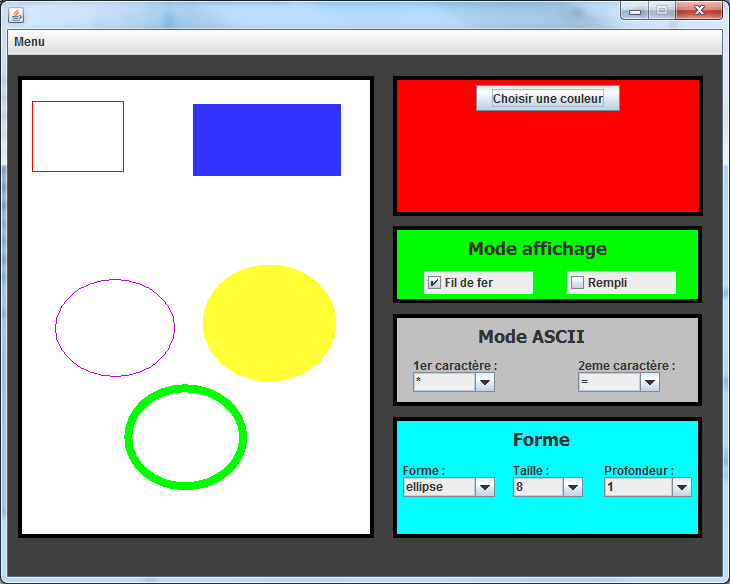
}

**else**

arg0.fillOval(x1, y1, x2-x1, y2-y1);

Exemple ici, si le mode est en fil de fer nous utiliserons l’objet « Graphics2D » pour construire la forme, sinon en mode rempli ça sera l’objet « Graphics ».

Et selon le type de la forme nous appelons soit Ellipse2D ou Rectangle2D en fil de fer et la méthode fillOval ou fillRect en mode rempli pour construire soit une ellipse ou soit un rectangle.

Rendu final :

Conclusion

Dans le cadre de notre projet, nous devions réaliser une application permettant de dessiner des formes toutes en prenant en compte le fait de rajouter des options d’effet.

Dans un premier temps, nous avons dû concevoir le diagramme de classe qui nous était utile pour la suite de notre projet pour avoir une vue global.

D’un point de vue humain, le travail en groupe est une bonne chose. En effet, cela permet de travailler en équipe et donc, de respecter les choix et idées de tous les membres afin de servir une cause commune. Aussi, la répartition des tâches permet un gain de temps sur le travail à effectuer. Également, le fait de découper le travail en module permet de se concentrer sur une partie précise où l’on va s’investir à fond dans l’intérêt du groupe. Le travail en groupe permet le partage des connaissances et chacun fait profiter les autres de son expérience ou de son expertise dans un domaine. Aussi, cela permet de **faire émerger de nouvelles idées** grâce au brainstorming mais ausside **résoudre** plus facilement les problèmes : la solution est recherchée à plusieurs, on n’est plus seul dans son coin à buter sur la difficulté.

Grâce à ce projet, nous avons pu perfectionner nos connaissances en langage JAVA. Malheureusement, le projet n’a pas était emmené à termes. Il reste quelques fonctionnalités qui n’ont pas était développé comme le mode ASCII ou la mise en place de la profondeur d’une forme. Bien que nous avons eu recours à de l’aide extérieur pour quelque explication, notamment sur les composants SWING permettant le dessin des formes, il a été assez dur de mener ce projet là où il en est.

David Guerry,

Maxime Hélias.