

MASTER1
Informatique

2 juin

2009

NGUYEN hong vu – Numéro de la carte d'étudiante : 10500181.

Rapport
Infographie

Rapport INFOGRAPHIE

Table des matières

A.	Guide d'utilisation de l'applet 2D :.....	3
A.1.	Menu déroulant :.....	3
A.2.	Les boutons :	4
B.	3D	6
	Classe Maillage .java.....	6



A. Guide d'utilisation de l'applet 2D :

But : Cette applet sert à dessiner des figures en deux dimensions. (2D)

- Le menu déroulant : pour sélectionner la figure à dessiner
- Les boutons : pour modifier la figure
- Le rectangle au centre : limite pour le fenêtrage

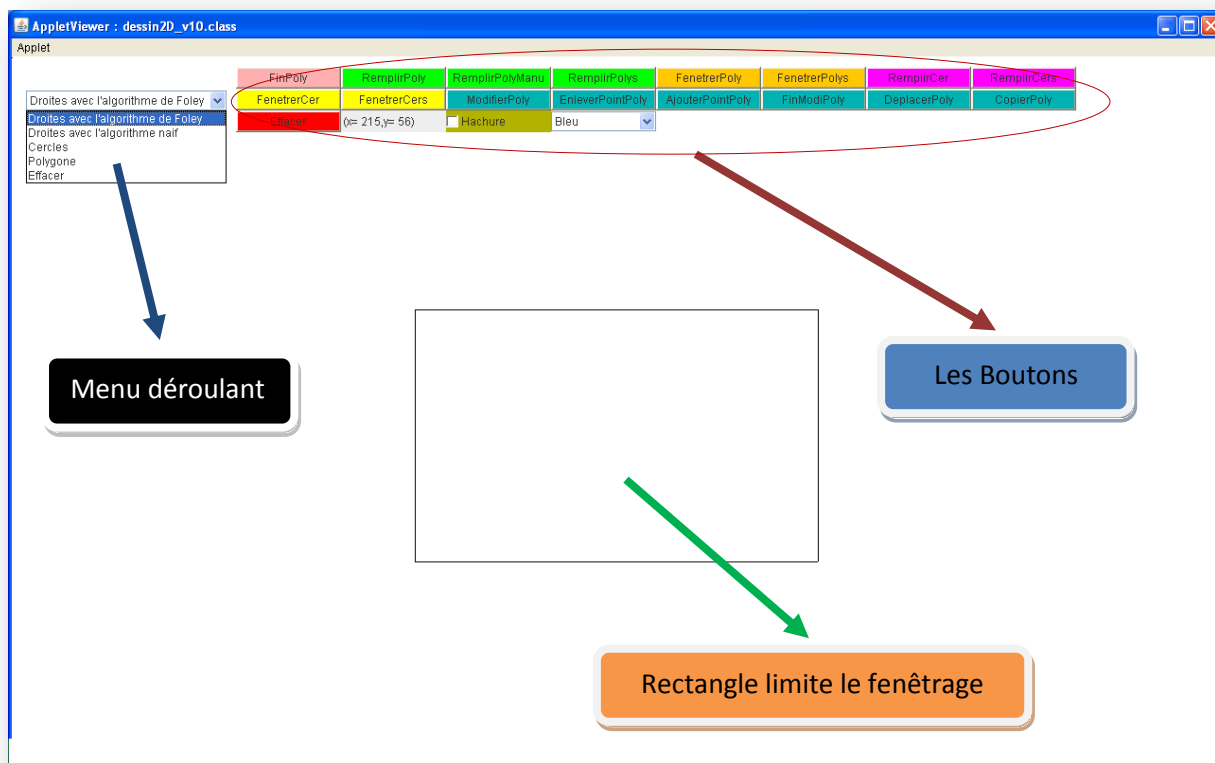


Fig1. Présentation générale de l'applet 2D

A.1. Menu déroulant :

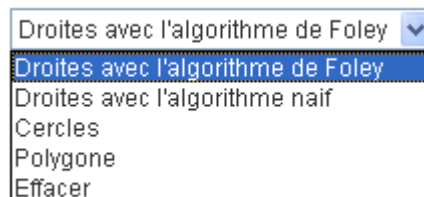


Fig2. Présentation du menu déroulant

– Droites avec l'algorithme de Foley :

 T racer une droite entre deux points avec l'algorithme de Foley.

Cliquer 2 fois de suite sur 2 endroits différents sur l'applet, une droite sera dessiner entre ces deux points.

– Droites avec l'algorithme naif :

 T racer une droite entre deux points avec l'algorithme naif.

Utiliser comme l'algorithme de Foley.

– Cercles :

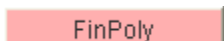
 T race un cercle avec un algorithme inspiré de Foley.

Cliquer 2 fois de suite sur 2 endroits différents de l'applet, le premier point sera le centre du cercle, le second détermine le rayon du cercle.

– Polygone :

 T race un polygone en utilisant l'algorithme de Foley .

Cliquer une suite des sommets souhaités pour former le polygone puis cliquer sur le bouton

« FinPoly » :  pour boucler le premier et le dernier point du polygone.

– Effacer :

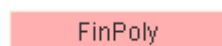
 Effacer le contenu de la fenêtre.

A.2. Les boutons :

Les boutons sont classés de gauche à droite et de haut vers bas.

FinPoly	RemplirPoly	RemplirPolyManu	RemplirPolys	FenetrerPoly	FenetrerPolys	RemplirCer	RemplirCers
FenetrerCer	FenetrerCers	ModifierPoly	EnleverPointPoly	AjouterPointPoly	DeplacerPoly	CopierPoly	SupprimerPoly
FinModiPoly	Effacer	(x= 156,y= 12)	Hachure	Bleu			

Fig3. Représentation les boutons.

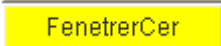
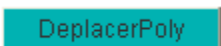
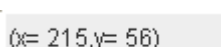
 : Boucler le dernier point et le premier point du polygone.

 : Remplir le dernier polygone tracé, avec la couleur sélectionnée.

 : Remplir manuellement un polygone : Cliquer sur un polygone dans l'applet pour le remplir avec la couleur sélectionnée.

 : Remplir tous les polygones présents sur l'applet avec la couleur sélectionnée



-  : Fenêtrer le dernier polygone.
-  : Fenêtrer tous les polygones existes.
-  : Remplir le dernier cercle dessiné avec la couleur sélectionnée
-  : Remplir tous les cercles existes sur l'applet avec la couleur sélectionnée.
-  : Fenêtrer le dernier cercle.
-  : Fenêtrer tous les cercles existes sur l'applet.
-  : Déplacer un sommet du dernier polygone dessiné. Cliquer la souris, l'applet sélectionne le sommet du polygone le plus proche du point où l'utilisateur lâche la souris.
-  : Enlever un sommet du dernier polygone dessiné. Cliquer la souris, l'applet sélectionne le sommet le plus proche du point où l'utilisateur lâche la souris.
-  : Ajouter un point saisi à la souris dans le dernier polygone dessiné. Cliquer la souris, l'applet sélectionne l'arrête la plus proche du point où l'utilisateur lâche la souris.
-  : Déplacer un polygone sélectionné. Cliquer sur un polygone pour le sélectionner, puis cliquer la 2^{ème} fois sur un autre endroit sur l'applet pour déplacer le polygone sélectionné.
-  : Copier un polygone sélectionné. Cliquer sur un polygone pour le sélectionner, puis cliquer la 2^{ème} fois sur un autre endroit sur l'applet pour créer une copie de cet polygone.
-  : Supprimer un polygone. Cliquer sur un polygone pour le supprimer
-  : Ce bouton met fin à toutes les actions « ModifierPoly », « EnleverPointPoly », « AjouterPointPoly », « DeplacerPoly », « CopierPoly », « SupprimerPoly ».
-  : Effacer le contenu de l'applet.
-  (x= 215,y= 56) : Ceci n'est pas un bouton, il sert pour savoir les cordonnées du curseur.
-  : Une case à cocher pour sélectionner option remplissage avec ou sans hachure.
-  : Les choix de couleurs remplissage.



B. 3D**Classe Maillage .java**

```

import java.util.Vector;
import java.awt.*;

public abstract class Maillage
{
    public Vector<Point3D> sommets;
    public Vector<Vector<Integer>> mailles;
    public Vector<Point2D> sommetsProjetes;

    public Vector<Point2D> maille2Polygone(int indiceMaille)
    {
        projeteSommets();

        Vector<Integer> mailleAProjeter = mailles.get(indiceMaille);
        Vector<Point2D> mailleProjete = new Vector<Point2D>();

        for(int i=0;i<mailleAProjeter.size();i++)
            mailleProjete.add(sommetsProjetes.get(mailleAProjeter.get(i)));

        return mailleProjete;
    }

    public Vector<Point2D> maille2Polygone(int indiceMaille,float z0,Point3D Ob)
    {
        projeteSommets(Ob.getZ()-z0,Ob);

        Vector<Integer> mailleAProjeter = mailles.get(indiceMaille);
        Vector<Point2D> mailleProjete = new Vector<Point2D>();

        for(int i=0;i<mailleAProjeter.size();i++)
            mailleProjete.add(sommetsProjetes.get(mailleAProjeter.get(i)));

        return mailleProjete;
    }

    public void projeteSommets()
    {
        sommetsProjetes.clear();

        for(int i=0;i<sommets.size();i++)
            sommetsProjetes.add(sommets.get(i).projetePoint3D());
    }

    public void projeteSommets(float d,Point3D Ob)
    {
        sommetsProjetes.clear();

        for(int i=0;i<sommets.size();i++)
            sommetsProjetes.add(sommets.get(i).projetePoint3D(d,Ob));
    }
}

```

