

PROJET TETRAEDRE [1. INTRODUCTION](#)

[2. MATHEMATIQUES 3D](#)

[3. DIAGRAMME DE CLASSE](#)

[4. CLASSE VECTEUR](#)

[5. CLASSE MATRICE](#)

[6. CLASSE TETRA APPLLET](#)

[7. COMPILATION ET EXECUTION APPLLET](#)

[8. AMELIORATION OU EVOLUTION POSSIBLE](#)

[9. CONCLUSION](#)

1. INTRODUCTION

Ce projet a pour but **la réalisation d'objet en 3d**, en l'occurrence un Tétraédre, pour la présentation sur un ou des sites Internet.

Cela nécessite aucune connaissance particulière ni en programmation JAVA, ni mathématique **pour son utilisation éventuelle** sur un site Internet/Intranet.

En revanche, **ces 2 connaissances** sont nécessaires pour sa modification ou sa compréhension:

☐☐☐ - **Géométrie analytique.**

☐☐☐ - **JAVA**

2. MATHEMATIQUES 3D:

L'outil mathématique est la projection sur un plan en l'occurrence l'écran, bien sûr.

Pour plus d'information que je ne développerais pas moi-même à ce sujet vous pouvez aller voir ce site universitaire:

<http://raphaello.univ-fcomte.fr/Ig/Visualisation/Visualisation.htm#Introduction>

3. DIAGRAMME DE CLASSE:

Le diagramme de classe a été réalisé en UML avec l'outil [Poseidon](#) :



- Class Vecteur: Création et utilisation de vecteurs
- Class Matrice: Création et utilisation de (4x4)
- Classe Tetra : Applet graphique de réalisation d'un Tétraédre

Pour un petit cours de Java vous pouvez télécharger un cours fait par Borland : [cours borland java.zip](#)

4. CLASSE VECTEUR:

```
Vecteur() {} Constructeur simple
Vecteur(double , double , double ) {} Constructeur void SetVecteur(Vecteur ) {} :
Affectation d'un vecteur à un autre void Add(Vecteur ) {} : Addition de 2 Vecteurs
void Substract(Vecteur ) {} Soustraction de 2 Vecteurs
void Multiply(double ) {} Multiplication par un réel d'un vecteur
void Divide(double ) {} Division par un réel non nul d'un vecteur
double GetX() 1ère coordonnée X
double GetY() 2nde coordonnée Y
double GetZ() 3ème coordonnée Z
void Initialize() {} : Initialisation
void SetX(double ) {} Attributions à X d'une valeur
void SetY(double ) {} Attributions à Y d'une valeur
void SetZ(double ) {} Attributions à Z d'une valeur
double Norme() Norme euclidienne d'un vecteur
void Normalize() {} Normalisation d'un vecteur
void Coherent() {} Rends les trois premières coordonnées du vecteur/ égale aux
coordonnées du point dans le monde 3D
void ProdVect(Vecteur Vecteur ) {} Produit Vectoriel de 2 vecteurs
double ProdScal(Vecteur vecteur) { return 0.0;} Produit scalaire de 2 vecteurs
void Multiply(Matrice Vecteur ) {} Produit Matrice par un vecteur
boolean determinant(Vecteur , Vecteur ) déterminant de 3 vecteur et retourne vrai si repère
direct faux sinon (utilisation savoir le sens de rotation par exemple)
```

5. CLASSE MATRICE

Constructeurs:

```
Matrice() {}
Matrice(double , double, double , double , double , double, double , double , double , double ,
, double , double , double , double , double , double) {}
```

// Methods

```
void SetMatrice(Matrice){}
void Set(double , double , double , double , double , double , double , double , double ,
double , double , double, double , double , double , double ) {} Attributions
void SetIJ(int , int , double ) {} Attribution element Matrice(I,J)
double GetIJ(int , int ,) {} Obtenir élément (i,j)
void Add(Matrice ) {} Addition de 2 Matrice
void Substract(Matrice ) {} Soustraction de 2 Matrices
void Multiply(double ) {} Multiplication par un réel
void Divide(double double0) {} Division par un réel non nul
void Scaling(Vecteur) {} Construction matrice d'homothétie
void Translation(Vecteur) {} Construction matrice de translation
void RotationX(double ) {} Rotation autour de l'axe x
void RotationY(double) {} Rotation autour de l'axe y
```

Projet Tetraedre

Écrit par Administrator

Lundi, 16 Février 2009 07:53 - Mis à jour Mercredi, 18 Février 2009 14:39

```
void RotationZ(double double0) { } Rotation autour de l'axe a  
void Multiply(Vecteur , Vecteur ) { } Multiplication par un vecteur  
void Multiply(Matrice , Matrice 1) { } Multiplication de 2 matrice
```

6. CLASSE TETRA

```
public Tetra() { } Constructeur  
public void init() { } Initialisation  
public void destroy() { } Destructeur  
public void update(Graphics) { } Rafraichissement du graphique avec bufferisation  
public void trace_arc(Graphics, int , Vecteur , int, Font) { } Tracé d'un arc  
public void paint(Graphics ) { } Tracé d'un graphique  
public boolean mouseDown(Event , int , int ) { } Click  
public boolean mouseMove(Event , int , int ) { return false;} MouseOver  
public void start() { } Début du Thread  
public void stop() { } Arrêt du Thread  
public void run() { }Exécution du Thread
```

7. COMPILATION ET EXECUTION APPLLET

Pour **pouvoir l'exécuter sur des JVM standard fourni par Microsoft** il faut le compiler avec une JDK <1.2 non compatible avec IE sauf en téléchargement un plus récente sur <http://java.sun.com> sinon il faut la compiler avec une JDK 1.1 comme celle dans le [download](#)

Pour l'exécution, il y a certain paramètre à entrer pour l'exécution:

```
<param name="bc" value="fond.jpg"> Image du fond d'appet  
<param name="centre_tetra" value="Moi"> Texte du centre du tétraèdre  
<param name="link1" value="cv.htm"> link1,link2... Liens relatifs  
<param name="link2" value="lien.htm">  
<param name="link3" value="download.htm">  
<param name="link4" value="autre.htm">  
<param name="titre1" value="CV"> titre1,titre2... Titres Sommet  
<param name="titre2" value="Liens">  
<param name="titre3" value="Download">  
<param name="titre4" value="Autres">  
<param name="sstitre1" value="Divers;Formation;Expérience;Général;"> sstitre1,sstitre2...  
SOus titre des sommets si absent _  
<param name="sstitre2" value="_;_;Divers;Langages">  
<param name="sstitre3" value="_;_;Autres;CV (PDF,Word)">  
<param name="sstitre4" value="_;_;Autres">  
<param name="statusbar" value="Bienvenue Chez Xavier Fontanet"> Titre barre de stratus du navigateur  
<param name="couleur1" value="255255255"> couleur RGB boules avant plan  
<param name="couleur2" value="051038017"> couleur RGB boules arrière plan
```

Projet Tetraedre

Écrit par Administrator

Lundi, 16 Février 2009 07:53 - Mis à jour Mercredi, 18 Février 2009 14:39

8 . AMELIORATION OU EVOLUTION POSSIBLE

- Paramétrisation supplémentaire: paramètre de l'ombre, nombre de sommet par exemple dans un cercle circonscrit (A venir)
- Zoom des sommets
- Effets : ombre présente mais plus estompé pr exemple
- D'autres si le cœur vous en dit je suis partant.

9. CONCLUSION

Ce petit exemple est free. Si le cœur vous en dit vous pouvez l'améliorer. [Contacter moi](#)

[Home](#)

[HAUT](#)