

Projet modélisation 3D

Equipe G3

Groupe G
Semestre 3
2021-2022

Victor Mougél
Clément Papin
Cyril Roux
Arthur Callens
Rémi Baran

Sommaire

- Fonctionnalités
- Éléments Techniques
 - Rotation
 - MVC
 - Tests
- Éléments d'organisation



Fonctionnalités

1) Fonctionnalités

- File Explorer
- Rotation, Translation, Homothétie
- Lumière & couleurs



Éléments Techniques

2) Éléments Techniques

Notre système de rotation et de transformation

```
/**
 * Permet d'effectuer la rotation de la matrice de rapport rotation
 * @param centre
 * @param rotation
 * @return La matrice après rotation
 */
public void rotationX(Point centre,double rotation,PlyModel model){
    matrice=calculCentre(centre, rotation, "rotationX");
    model.actualizePoints();
}

public void rotationY(Point centre,double rotation,PlyModel model) {
    matrice=calculCentre(centre, rotation, "rotationY");
    model.actualizePoints();
}
```

```
/**
 * Ramene la matrice au centre, effectue une multiplication suivant le type de transformation souh  te et la renvoie au point d'origine
 * tout ca pour simplifier les calculs
 * @param centre le centre de la matrice
 * @param valeur de la transformation (rapport d'omoth  tie ou rotation);
 * @param typeTransformation le type de transformation, homoth  tie ou rotation
 * @return la matrice apr  s transformation
 */
private Matrice calculCentre(Point centre,double valeur,String typeTransformation){
    Point opposit=new Point(-centre.getCoX(),-centre.getCoY(),-centre.getCoZ());

    Matrice versOrigine=transformations.translationMatrice(opposit);
    Matrice transformation= typeTransformation.equals("homoth  tie") ? transformations.homoth  tieMatrice(valeur) : typeTransformation.equals("rotationX") ? transformations.rotationMatriceX(valeur) : transformations.rotationMatriceY(valeur);
    Matrice versCentre=transformations.translationMatrice(centre);

    versOrigine=versOrigine.productMatrice(matrice);
    transformation=transformation.productMatrice(versOrigine);
    versCentre=versCentre.productMatrice(transformation);

    return versCentre;
}
```

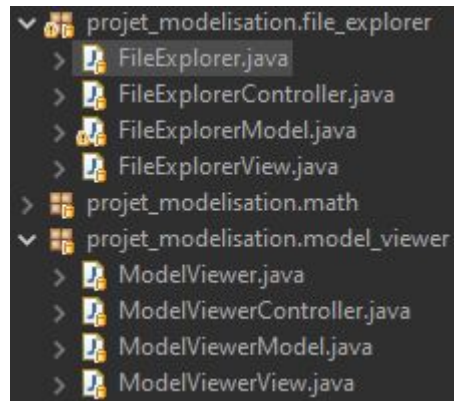
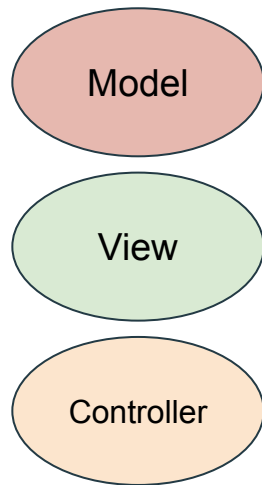
2) Éléments Techniques

Fonction de calcul de la matrice de rotation

```
public Matrice rotationMatriceX(double k) {  
    return new Matrice(new double[][] { {1,0,0,0}, {0,Math.cos(k),Math.sin(k),0}, {0,-Math.sin(k),Math.cos(k),0}, {0,0,0,1}});  
}  
  
public Matrice rotationMatriceY(double k) {  
    return new Matrice(new double[][] { {Math.cos(k),0,-Math.sin(k),0}, {0,1,0,0}, {Math.sin(k),0,Math.cos(k),0}, {0,0,0,1}});  
}
```

2) Éléments Techniques

L'utilisation du MVC dans notre projet



2) Éléments Techniques

Les différents tests que nous avons utilisés

2 types de test :



```
graph TD; A[2 types de test :] --> B[Test sur les calculs]; A --> C[Test sur les fichiers];
```

Tests sur les calculs

```
public class TestModeleMatrice
```

```
public class TestMatrice
```

Tests sur les fichiers

```
public class TestPly
```



Éléments d'organisation

3) Éléments d'organisation

- Début projet -> Livrable 1
- Livrable 1 -> Livrable 2



Début projet -> Livrable 1



Partie Maths Victor / Clément	Partie MVC / Affichage Cyril / Rémi / Arthur
matrice / calcul matricielle / rotation / translation / homothétie	lecture fichier / affichage / modèle / vue / contrôleur / fichier ply / ...



Livable 1 -> Livable 2



Victor : rotation auto
Cyril : clean code
Rémi et Arthur : les test
Clément : la lumière



Merci de votre attention