Projet de Modélisation 3D

Projet présenté par Julien Lalloyer,
Mathéo Sinquin, Christopher Potier

Sommaire

C

01

Présentation du sujet

Démonstration

02

La lumière

Découvrez tous nos calcules de lumière

J

Comment est composé la lumière ?

Produit Scalaire :

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

```java
public class ProduitScalaire {

    /** La matrice. */
    protected Matrix matrice;
```

```java
    */
    public ProduitScalaire(Matrix matrice) {
        this.matrice = matrice;
    }
```

```java
public ProduitScalaire(Vertex sommet1, Vertex sommet2) {
    matrice = new Matrix(1,1);
    matrice.add(sommet2.getX() - sommet1.getX(), sommet2.getY() - sommet1.getY(), sommet2.getZ() - sommet1.getZ(), 1);
}
```

Comment est composé la lumière ?

Produit Scalaire :

$$u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \text{ et } v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

```
public void prodScal(Matrix matrice) {
    Matrix res;
    if(verifProd(matrice)){
        res = calcScal(matrice);
    }else {
        res = new Matrix(1,1);
    }
    this.matrice  = res;
}
```

```
public boolean verifProd(Matrix matrice) {

    return this.matrice.getLength() == matrice.getLength();
}
```

```
public Matrix calcScal(Matrix matrice) {
    Matrix res = new Matrix(1,1);
    double coordX = this.matrice.getY(0) * matrice.getZ(0) - this.matrice.getZ(0) * matrice.getY(0);
    double coordY = this.matrice.getZ(0) * matrice.getX(0) - this.matrice.getX(0) * matrice.getZ(0);
    double coordZ = this.matrice.getX(0) * matrice.getY(0) - this.matrice.getY(0) * matrice.getX(0);
    res.add(coordX, coordY, coordZ, 1);
    return res;
}
```

Comment est composé la lumière ?

Produit Scalaire :

Soient $u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$ et $v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$. On définit le *produit vectoriel*[1] de u et v par

$$u \wedge v = \begin{pmatrix} \det \begin{pmatrix} u_2 & v_2 \\ u_3 & v_3 \end{pmatrix} \\ \det \begin{pmatrix} u_3 & v_3 \\ u_1 & v_1 \end{pmatrix} \\ \det \begin{pmatrix} u_1 & v_1 \\ u_2 & v_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{pmatrix}$$

(0)

Formule du produit vectoriel

Comment est composé la lumière ?

Produit Vectoriel Unitaire :

```
public class ProdVectoUni {
protected ProduitScalaire produitScalaire;
protected Matrix matrice;
protected double norme;
private double coordX;
private double coordY;
private double coordZ;
```

```
public ProdVectoUni(ProduitScalaire produitScal){
    produitScalaire = produitScal;
    coordX = produitScalaire.getMatrice().getX(0);
    coordY = produitScalaire.getMatrice().getY(0);
    coordZ = produitScalaire.getMatrice().getZ(0);
    norme = Math.sqrt(coordX*coordX + coordY*coordY +coordZ*coordZ);
    matrice = new Matrix(1,1);
    matrice.add(coordX/norme, coordY/norme, coordZ/norme, 1);
}
```

Afin de savoir l'orientation de la face

Comment est composé la lumière ?

Calcule Color :

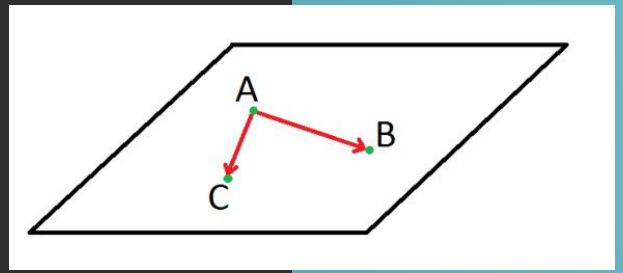
```
public class CalculColor {
Color couleur;
Matrix vecteurlumiere;
```

```
public CalculColor(Matrix positionlumiere, Color color) {
    ProduitScalaire produitScalaire = new ProduitScalaire(positionlumiere);
    ProdVectoUni produitVectoUni =  new ProdVectoUni(produitScalaire);
    this.vecteurlumiere = produitVectoUni.getNorme();
    couleur = color;
}
```

Comment est composé la lumière ?

Calcule Color :

```
public Color getColor(Face face) {
    ArrayList<Vertex> listsommet = (ArrayList<Vertex>) face.getVertex();
    if(face.getNbVertex()== 3) {
        ProduitScalaire produitScalaire = new ProduitScalaire(listsommet.get(0), listsommet.get(1));
        produitScalaire.prodScal(new ProduitScalaire(listsommet.get(0), listsommet.get(2)).getMatrice());
        ProdVectoUni prodVectoUni = new ProdVectoUni(produitScalaire);
        // la il reste a faire le calcule avec la couleur
        double coef = calcfactlumiere(prodVectoUni);
        double rouge = face.getRed();
        double vert = face.getGreen();
        double bleu = face.getBlue();
        if(coef > 0.0) {
            couleur = new Color((rouge * coef), (vert * coef), (bleu* coef), 1.0);
        }else {
                couleur = new Color(0.0,0.0,0.0,1.0);
            }
    }
    return couleur;
}
```



Comment est composé la lumière ?

Calcule Color :

```
private double calcfactlumiere(ProdVectoUni produitVectoUni) {
    double coord_x = produitVectoUni.getNormeX() * vecteurlumiere.getX(0);
    double coord_y =produitVectoUni.getNormeY() * vecteurlumiere.getY(0) ;
    double coord_z =produitVectoUni.getNormeZ() * vecteurlumiere.getZ(0);
    return coord_x+coord_y+coord_z;

}
```

03

Calcul Matriciel

Découvrez nos calculs matriciels en détail

Comment est composé la matrice ?

```java
public class Matrix {

    private int idx = 0, length;

    private double[] xCoordinate ,| yCoordinate , zCoordinate , vector;
```

```java
public Matrix(int nbVertex, int nbFaces) {
    this.length = nbVertex;
    this.xCoordinate = new double[length];
    this.yCoordinate =new double[length];
    this.zCoordinate =new double[length];
    this.vector =new double[length];
}
```

$$A = \begin{pmatrix} 1 & 2 & -1 & 2 \\ 2 & 5 & -2 & 2 \\ 1 & 2 & 1 & 2 \end{pmatrix}$$

```java
public void add(double xCoordinate, double yCoordinate, double zCoordinate, double vector) {
    this.xCoordinate[idx] = xCoordinate;
    this.yCoordinate[idx] = yCoordinate;
    this.zCoordinate[idx] = zCoordinate;
    this.vector[idx] = vector;
    idx ++;
}

public void add(Vertex vertex) {
    this.add(vertex.getX(),vertex.getY(),vertex.getZ(), 1);
}
```

Comment est composé la matrice ?

```java
public Graph(int nbFace, List<Face> faces, Matrix matrix, List<String> listOfFaces, String author) {
    this.originalMatrix = matrix;
    this.matrix = matrix;
    this.nbFaces = nbFace;
    this.faces = faces;
    this.listOfFaces = listOfFaces;
    this.author = author;
    this.color = this.faces.get(0).getColor();
}
```

```java
public UpdateGraph(int nbFaces, List<Face> faces, Matrix matrix, List<String> li
    super(nbFaces, faces, matrix, listOfFaces, author);
    this.original=new Graph(nbFaces, faces, matrix, listOfFaces, author);
}

/**
 * Update the graph with a new matrix
 *
 * @param matrix the new matrix
 */
public void update(Matrix matrix) {
    this.faces=this.original.faces;
    this.matrix=this.original.matrix;
    this.listOfFaces=this.original.listOfFaces;
    this.matrix = matrix;
    faces.clear();
    vertices.clear();
    for(int i = 0; i<matrix.getLength(); i++) {
        vertices.add(new Vertex(matrix.getX(i),matrix.getY(i),matrix.getZ(i)));
    }
    readFace();
    notifyObserver();
}
```

Comment fonctionne la matrice ?

```java
public class Homothety extends Mouvement {

    /** the current matrix */
    private Matrix currentMatrix;
```

```java
private void homothetieAction(double coef) {
    homothetie = new Homothety(graphe.getMatrix());
    homothetie.mouvement(coef);
    graphe.update(homothetie.getCurrentMatrix());
}
```

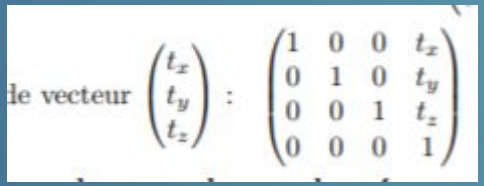
```java
@Override
public Matrix mouvement(double sensibility) {
    Matrix newMatrice = new Matrix(4, 4);
    newMatrice.add( sensibility,0.0, 0.0, 0.0);
    newMatrice.add( 0.0, sensibility, 0.0, 0.0);
    newMatrice.add(0.0, 0.0, sensibility, 0.0);
    newMatrice.add(0.0, 0.0, 0.0, 1.0);
    this.currentMatrix = multipliMatrix(newMatrice);

    return currentMatrix;
}
```

```java
buttonHomothetieUp.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        homothetieAction(1.2);
    }
});
```

$$\begin{pmatrix} k & 0 & 0 & 0 \\ 0 & k & 0 & 0 \\ 0 & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Comment fonctionne la matrice ?

```java
public Translation(Matrix matrix) {
    currentMatrix = matrix;
}
```




```java
public  Matrix addToMatrice(double coordx, double coordy , double coordz) {
    Matrix matriceConverted = new Matrix(currentMatrix.getLength(),currentMatrix.getLength());
    for(int i = 0 ; i < currentMatrix.getLength(); i++) {
        xCoordinate =  currentMatrix.getX(i) + coordx;
        yCoordinate =  currentMatrix.getY(i) + coordy;
        zCoordinate =  currentMatrix.getZ(i) + coordz;
        matriceConverted .add(xCoordinate, yCoordinate, zCoordinate, currentMatrix.getV(i));
        xCoordinate = 0.0;
        yCoordinate = 0.0;
        zCoordinate = 0.0;
    }
    currentMatrix = matriceConverted;
    return currentMatrix;
}

/**
 * Do the translation
 *
 * @param matrix the current matrix
 * @return the matrix after the translation
 */
public Matrix translate(Matrix matrice) {
    this.addToMatrice(matrice.getX(0), matrice.getY(0),matrice.getZ(0));
    return this.getMcourante();
```

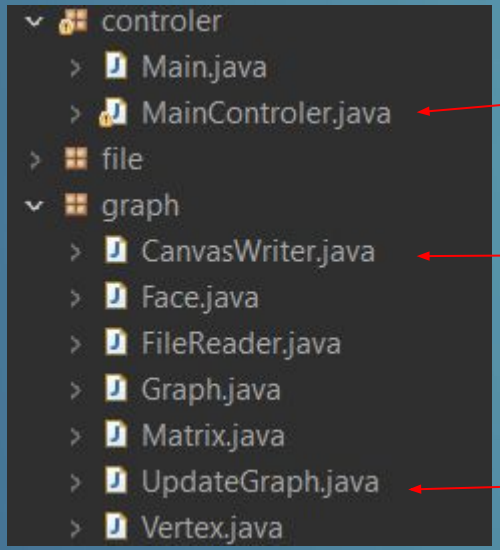
```java
buttonTranslateRight.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        if(currentFile != null && currentFile.isFile()) {
            translateAction(0.5,0.0,0.0);
        }
    }
});
```

```java
private void translateAction(double xCoordinate, double yCoordinate, double zCoordinate) {
    translation = new Translation(graphe.getMatrix());
    translation.translate(new Matrix(1, xCoordinate , yCoordinate, zCoordinate ,1.0));
    graphe.update(translation.getMcourante());

}
```

04

Architecture MVC

Comment est composé la matrice ?



controleur

Vue

Modéle

05

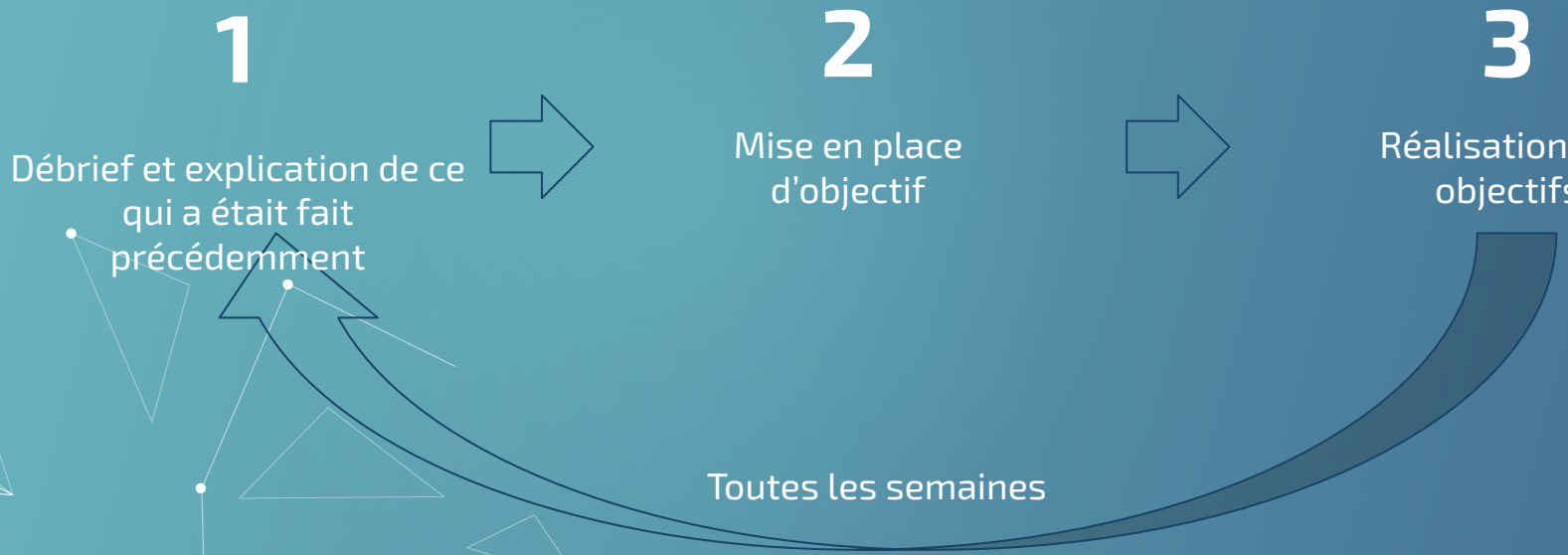
Organisation

Un travail organisé pour un programme de qualité

J

Modélisation

Organisation d'une réunion de projet



J

06

Conclusion

Tout connaître sur notre organisation

J

Conclusion

Retour sur les difficultées rencontrées:

Le MVC

Vue en tranche que
l'on a pas pus finir

Controleur Horloge

Conclusion

Retour sur expérience équipe et personnelle

Personnelle :

Julien	Christopher	Mathéo
- Rythme très satisfaisant - Résultat satisfaisant - J'ai adoré travailler dessus	- Le sujet était très sympathique. - Bonne symbiose dans le groupe . - Assez fier de ce que le groupe a réussi à faire	- Très satisfait de ce groupe et du travail - Dès qu'il y avait un blocage les autres pouvaient aider - Bonne répartition des tâches

Conclusion

Apport du projet dans le semestre

Gestion de Projet :

Gestion d'un groupe

Meilleur organisation

Informatique :

Approfondissement des connaissances

Bon rappel sur certaines méthodes

Mathématique :

Mise en pratique des cours

Amélioration dans ce domaine

Conclusion

Retour sur expérience équipe et personnelle

Groupe :

Bonne Ambiance

Bonne Motivation dès le début

Rythme très soutenue et efficace

Super agréable, sur les créneaux de Projet

Très peu de moment sans travail

Merci de nous avoir écoutés

Avez vous des questions ?

Projet présenté par Julien Lalloyer,
Mathéo Sinquin, Christopher Potier