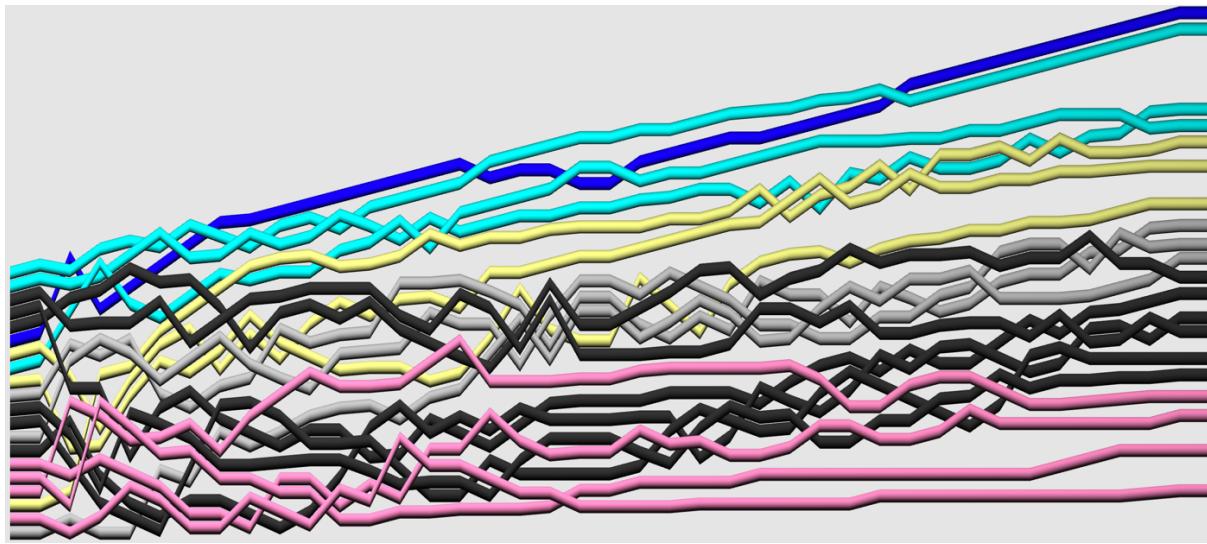


Rapport projet IGSD



Introduction :

Ce projet a pour but de réaliser un affichage OpenGL depuis un programme C++ qui affiche un gapchart un 3D. Ce gapchart représente la progression des équipes de la Premier League 2018-2019.

Lectures des données :

```

void loadData(string fn, vector<int> &pts, vector<int> &ranks) {
    ifstream file (fn); // declare file stream: http://www.cplusplus.com
    string value;
    // int count = 0;

    //Nombre d'équipes
    for(int j = 0; j < 20; j++){

        //Lis le nom de l'équipe
        getline ( file, value, ',' ); // read a string until next
        string teamName = string( value, 0, value.length() );

        //Nombre de journées
        for(int i = 0; i < 41; i++){
            //Le rang
            getline ( file, value, ',' );
            string rank = string( value, 0, value.length() );
            int r = stoi(rank, NULL);
            ranks[j*41+i] = r;

            //Les points
            getline ( file, value, ',' );
            string points = string( value, 0, value.length() );
            int p = stoi(points, NULL);
            pts[j*41+i] = p;

            //Match --> nom de l'équipe à domicile
            getline ( file, value, ',' );
            string homeTeam = string( value, 0, value.length() );

            //Nombre de buts pour l'équipe à domicile
            getline ( file, value, ',' );
            string homeGoal = string( value, 0, value.length() );
            int hGoal = stoi(homeGoal, NULL);

            //Nombre de buts pour l'équipe à l'extérieur
            getline ( file, value, ',' );
            string awayGoal = string( value, 0, value.length() );
            int aGoal = stoi(awayGoal, NULL);

            //Match --> nom de l'équipe à l'extérieur
            getline ( file, value, ',' );
            string awayTeam = string( value, 0, value.length() );
        }
        getline ( file, value, '\n' );
    }

    //Man City, 5, 0, 77,-1,-1,77, 5, Arsenal, 0, 2, Man City, 6, 0, 77, 5, Man City, 3, 0, 77, 5, Arsenal, 0, 2, Man City, 2, 0, Brighton, 0, 28, Liverpool, 0, 0, Man City, 0, 23, Man City, 5, 0, Fulham, 1, 16, Cardiff, 0, 5, Man City, 0, 19, Man City, 2, 0, Brighton, 0, 28, Liverpool, 1, 2, Man City, 0, 32, Man City, 3, 1, Man United, 0, 35, West Ham, 0, 1, Man City, 0, 29, Man City, 6, 1, Southampton, 0, 32, Man City, 3, 1, Man United, 0, 35, West Ham, 0, 1, Man City, 0, 38, Southampton, 3, 1, Bournemouth, 0, 41, Watford, 1, 2, Man City, 0, 31, Chelsea, 0, 0, Man City, 0, 44, Man City, 3, 2, Everton, 0, 14, Man City, 2, 0, Crystal Palace, 0, 24, Leicester, 0, 1, Man City, 0, 47, Southampton, 1, 2, Man City, 0, 49, Bournemouth, 0, 1, Man City, 1, 50, Man City, 3, 0, Wolves, 1, 56, Huddersfield, 0, 3, Man City, 1, 56, Newcastle, 2, 0, Man City, 1, 59, Man City, 3, 1, Arsenal, 1, 62, Man City, 6, 0, Chelsea, 1, 65, Everton, 0, 2, Man City, 1, 68, Man City, 0, 0, West Ham, 0, 71, Bournemouth, 0, 1, Man City, 0, 74, Man City, 3, 1, Watford, 0, 77, Man United, 0, 2, Man City, 0, 80, Fulham, 0, 2, Man City, 0, 83, Man City, 2, 0, Cardiff, 0, 74, Liverpool, 0, 77, 5, Man City, 0, 84, Bournemouth, 0, 2, Man City, 0, 87, Watford, 0, 77, 5, Man City, 0, 89, Fulham, 0, 1, Man City, 0, 90, Leicester, 0, 1, Bournemouth, 0, 92, Burnley, 0, 1, Man City, 0, 95, Man City, 1, 0, Leicester, 0, 96, Brighton, 0, 1, Man City, 0, 98, 77,-1,-1,77, Liverpool, 0, 77, 5, Liverpool, 1, 0, Liverpool, 0, 9, Liverpool, 1, 0, Brighton, 0, 12, Leicester, 1, 2, Liverpool, 1, 15, Tottenham, 1, 2, Liverpool, 0, 18, Liverpool, 0, 3, Southampton, 1, 19, Chelsea, 0, 12, Liverpool, 0, 20, Liverpool, 1, 23, Liverpool, 0, 26, Liverpool, 0, 4, Cardiff, 0, 27, Arsenal, 0, 1, Liverpool, 0, 21, Watford, 0, 23, Cardiff, 0, 3, Liverpool, 0, 27, Liverpool, 0, 30, Burnley, 1, 3, Liverpool, 0, 42, Bournemouth, 0, 4, Liverpool, 0, 45, Liverpool, 3, 1, Man United, 0, 48, Wolves, 0, 2, Liverpool, 0, 51, Liverpool, 0, 4, Newcastle, 0, 51, 77,-1,-1,77, 54, Liverpool, 5, 1, Arsenal, 0, 54, Man City, 2, 1, Liverpool, 0, 51, Brighton, 0, 6, Liverpool, 0, 66, Liverpool, 0, 4, Crystal Palace, 0, 61, Liverpool, 1, 1, Leicester, 0, 62, West Ham, 0, 51, Liverpool, 0, 65, Liverpool, 0, 69, Bournemouth, 0, 61, Man United, 0, 73, Liverpool, 0, 73, Liverpool, 0, 73, Watford, 0, 74, Everton, 0, 8, Liverpool, 0, 74, Burnley, 0, 2, Burnley, 0, 75, Fulham, 0, 2, Liverpool, 1, 76, Liverpool, 2, 1, Tottenham, 1, 82, Southampton, 1, 3, Liverpool, 1, 85, Liverpool, 2, 0, Chelsea, 1, 88, Cardiff, 0, 2, Liverpool, 1, 91, Liverpool, 0, 5, Huddersfield, 1, 94, Newcastle, 2, 3, Liverpool, 1, 97, Liverpool, 2, 0, Wolves, 1, 97, 77,-1,-1,77, Chelsea, 0, 77,-1,-1,77, 54, Huddersfield, 0, 3, Chelsea, 2, 0, Chelsea, 0, 9, Newcastle, 1, 2, Chelsea, 1, 12, Liverpool, 0, 77, 5, Liverpool, 1, 23, Liverpool, 0, 10, Chelsea, 0, 12, Liverpool, 0, 18, Liverpool, 1, 20, Southampton, 0, 3, Chelsea, 2, 21, Chelsea, 2, 2, Man United, 0, 24, Burnley, 0, 6, Chelsea, 1, 25, Chelsea, 3, 1, Chelsea, 2, 28, Chelsea, 0, 0, Everton, 3, 28, Tottenham, 1, 3, Chelsea, 2, 31, Chelsea, 2, 0, Fulham, 3, 31, Wolves, 2, 1, Chelsea, 3, 34, Chelsea, 2, 0, Man City, 3, 37, Brighton, 2, 1, Chelsea, 3, 37, Chelsea, 0, 1, Leicester, 0, 40, Watford, 1, 2, Chelsea, 3, 37, Chelsea, 0, 1, Bournemouth, 0, 43, Crystal Palace, 0, 40, Watford, 1, 2, Chelsea, 3, 47, Arsenal, 2, 0, Chelsea, 4, 47, Bournemouth, 0, 4, Chelsea, 3, 50, Chelsea, 0, 5, Huddersfield, 5, 50, Man City, 6, 0, Chelsea, 3, 53, Chelsea, 0, 3, Brighton, 4, 56, Chelsea, 2, 0, Tottenham, 3, 59, Fulham, 1, 2, Chelsea, 4, 60, Chelsea, 1, 1, Wolves, 4, 60, Everton, 2, 0, Chelsea, 4, 63, Cardiff, 1, 2, Chelsea, 3, 66, Chelsea, 2, 0, West Ham, 4, 66, Liverpool, 0, 2, Chelsea, 3, 67, Chelsea, 2, 0, Burnley, 3, 68, Man United, 1, 1, Chelsea, 2, 71, Chelsea, 3, 0, Watford, 2, 72, Leicester, 0, 77, 5, Chelsea, 3, 77, 54, Liverpool, 0, 77, 54, Liverpool, 1, 77, 54, Liverpool, 2, 77, Tottenham, 1, 77,-1,-1,77, 77, 54, Newcastle, 1, 2, Tottenham, 4, 6, Tottenham, 1, 3, Fulham, 1, 0, Man United, 0, 3, Tottenham, 4, 9, Watford, 2, 1, Tottenham, 5, 9, Tottenham, 1, 2, Liverpool, 4, 12, Brighton, 2, 2, Tottenham, 3, 15, Huddersfield, 0, 2, Tottenham, 4, 18, Tottenham, 1, 0, Cardiff, 0, 21, West Ham, 0, 1, Tottenham, 4, 21, Tottenham, 0, 1, Man City, 3, 24, Wolves, 2, 3, Tottenham, 4, 27, Crystal Palace, 0, 24, Watford, 0, 27, Tottenham, 1, 0, Burnley, 2, 42, Everton, 2, 6, Tottenham, 3, 1, Southampton, 2, 31, Leicester, 0, 41, Tottenham, 2, 39, Tottenham, 1, 0, Burnley, 2, 42, Everton, 2, 6, Tottenham, 1, 45, Tottenham, 5, 0, Bournemouth, 1, 45, 77,-1,-1,77, 2, 45, Tottenham, 1, 3, Wolves, 2, 48, Cardiff, 0, 3, Tottenham, 2, 48, Tottenham, 0, 1, Man United, 2, 51, Fulham, 1, 2, Tottenham, 2, 54, Tottenham, 1, 2, Watford, 2, 57, Tottenham, 1, 0, Newcastle,

```

Pour la lecture des données, il m'a fallu implanter la fonction *loadData(...)* qui pour paramètres, un string qui est le nom de fichier à lire, et deux tableaux à remplir, l'un avec le rang des différentes équipes au cours du championnat et l'autre avec leurs points. Cette partie a été relativement facile et sans problème, mon plus gros challenge a été de bien séparer les différents éléments du document (nom de l'équipe, rang, points, etc...).

Création des points pour l'affichage et la visualisation :

```

//First Triangle
//Top left
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18] = j/41.; //X
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 1] = hauteur + 1*r*cos(theta); //Y
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 2] = r*sin(theta)+ r2*sin(l*M_PI/8); //Z

//Bottom left
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 3] = j/41.; //X
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 4] = hauteur + 1*r*cos((theta+(M_PI/16))); //Y
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 5] = r*sin((theta+(M_PI/16)))+r2*sin(l*M_PI/8); //Z

//Top right
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 6] = (j+1)/41.; //X
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 7] = hauteurProchain + 1*r*cos(theta); //Y
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 8] = r*sin(theta)+r2*sin((l*M_PI/8)); //Z

//Second Triangle
//Top right
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 9] = (j+1)/41.; //X
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 10] = hauteurProchain + 1*r*cos(theta); //Y
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 11] = r*sin(theta)+r2*sin((l*M_PI/8)); //Z

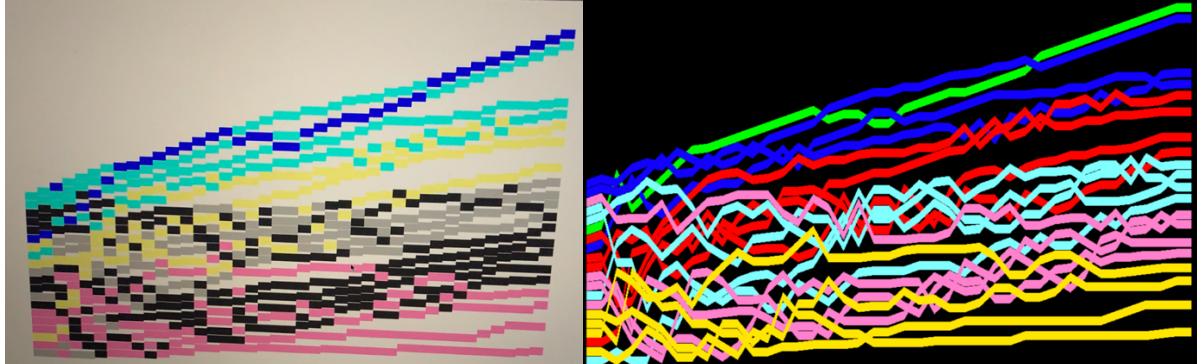
//Bottom left
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 12] = j/41.; //X
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 13] = hauteur + 1*r*cos((theta+(M_PI/16))); //Y
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 14] = r*sin((theta+(M_PI/16)))+r2*sin(l*M_PI/8); //Z

//Bottom right
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 15] = (j+1)/41.; //X
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 16] = hauteurProchain + 1*r*cos((theta+(M_PI/16))); //Y
gVertexBufferData[i * 41 *taille+ j * taille+ k*18*8+l*8*18+ 17] = r*sin((theta+(M_PI/16)))+r2*sin(l*M_PI/8); //Z

```

Figure 1. Image du calcul des données (3D)

Cette partie a été particulièrement difficile, en effet, il m'a fallu, tout d'abord calculer les points pour un affichage 2D, ce qui n'a pas été trop difficile, vu que toutes les indications étaient données dans le sujet. Après avoir réussi à afficher que des rectangles, il a juste fallu les relier entre eux. Pour cela, il a juste fallu calculer les points d'un rectangle à la position t+1 en terme de rang et de points.



(L'image de gauche n'est pas la mienne, j'ai oublié de prendre une photo).

Maintenant que j'ai réussi à afficher en 2D, il ne me restait plus que la 3D, pour cela, j'ai dû couper mes rectangles en plusieurs petits rectangles pour ensuite leur donner du

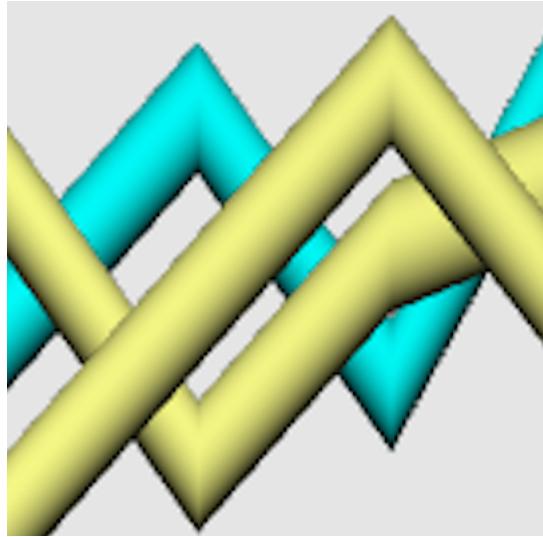


Figure 2. Impression de 3D

“volume”. En gros, il faut juste changer la coordonnée z de certains vertex pour obtenir un demi-cylindre. De plus, on remarque sur la photo (fig.2) qu'il y a un effet de lumière et d'ombrage. J'ai, en effet, créer une lumière ambiante dans les shaders, à laquelle j'ai ensuite

combiné les couleurs associées aux vertex pour obtenir le résultat ci-dessus.

```
//Lumiere ambiente
float ambientStrength = 0.1;
vec3 ambient = ambientStrength * vec3(1.0);

//Lumiere diffusee
vec3 lightDir = normalize(lightPos - FragPos);
float diff = max(dot(normal, lightDir), 0.0);
vec3 diffuse = diff * vec3(1.0);

//Rendu final de la couleur plus la lumiere
vec3 finalColor = (ambient + diffuse) * colort;
```

Interactions :

Concernant les interactions, j'ai juste implémenté une fonction qui me permet de déplacer mon graphe si jamais on appuyait sur les flèches du clavier.

```
//Angle de rotation dépendant de l'axe
float rotAngleY = 0.0f;
float rotAngleX = 0.0f;

void key_callback(GLFWwindow* window, int key, int scancode, int action, int mods){
    if (key == GLFW_KEY_LEFT && action == GLFW_PRESS){
        //Augmente l'angle de rotation si on appuie sur "LEFT"
        rotAngleY = (rotAngleY+M_PI/100);
    } else if (key == GLFW_KEY_RIGHT && action == GLFW_PRESS){
        //Augmente l'angle de rotation si on appuie sur "RIGHT"
        rotAngleY = (rotAngleY-M_PI/100);
    } else if (key == GLFW_KEY_UP && action == GLFW_PRESS){
        //Augmente l'angle de rotation si on appuie sur "UP"
        rotAngleX = (rotAngleX+M_PI/100);
    } else if (key == GLFW_KEY_DOWN && action == GLFW_PRESS){
        //Augmente l'angle de rotation si on appuie sur "DOWN"
        rotAngleX = (rotAngleX-M_PI/100);
    }

    //On multiplie la matrice pour garder les rotations sur les deux axes
    modelMatrix *= glm::rotate(modelMatrix, rotAngleY, vec3(0.0f, 1.0f, 0.0f));
    modelMatrix *= glm::rotate(modelMatrix, rotAngleX, vec3(1.0f, 0.0f, 0.0f));
}
```

C'est la fonction `key_callback(...)` qui s'en occupe. On peut observer que j'ai déclaré deux angles `rotAngleY` et `rotAngleX` auxquels je vais ajouter un certain offset. Ces angles seront ensuite utilisés dans la matrice `modelMatrix`, à laquelle on applique une rotation sur l'axe X ou Y dépendant de comment on veut faire bouger notre gapchart.

