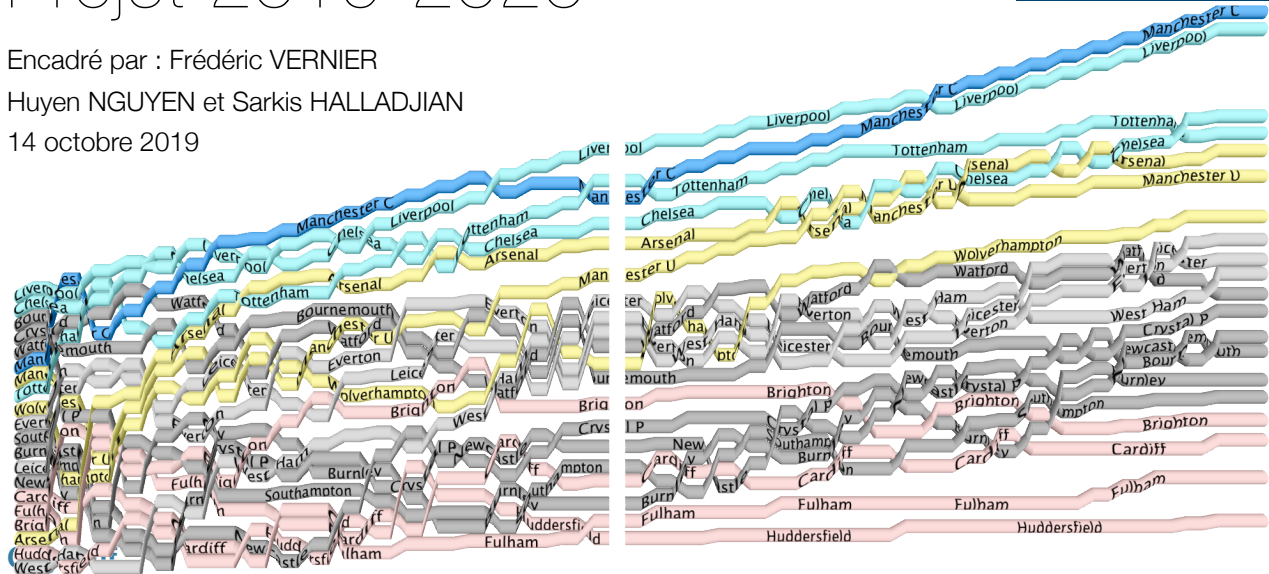


Projet 2019-2020

Encadré par : Frédéric VERNIER

Huyen NGUYEN et Sarkis HALLADJIAN

14 octobre 2019



L'objectif de ce projet est de réaliser un affichage OpenGL depuis un programme C++ qui affiche des gapchart en trois dimensions. Un gapchart représente un classements et des nombres de points qui évoluent dans le temps. Il peut s'agir du classement de Shanghai des universités ou d'un championnat de sport.

Données (<http://vernier.frederic.free.fr/Teaching/IGSD/DataProjet2019.zip>)

Vous disposerez d'un fichiers de données (rankspts.csv) qui représente les classements et les nombres de points de 20 équipes lors d'un championnat de 38 journées. Le fichier commence par :

```
Manchester C, 5, 0, ??,-1,-1,??,5, 3, Arsenal,0, 2,
Manchester C, 0, 6, Manchester C,6, 1,
Huddersfield,...
```

Donc l'équipe "Manchester C" a commencé 6ième (rang 5) avec 0 points puis a été 6ième (rang 5) avec 3 points. A la troisième journée il est devenu 1er (rang 0) avec 6 points, etc. Les données entre deux paires de classements/points correspondent au match qui a amené cette équipe à ce classement. Rien n'explique le classement initial (??,-1,-1,??) Mais le 2ième jour (rang 5 avec 3 pts) s'explique par une victoire 2-0 contre Arsenal, etc.

Les six premières colonnes sont artificielles puisque les 20 équipes commencent avec 0 points et avec le même classement qu'à la colonne 8 (2 fois le chiffre 5 qui correspond à la 6ième place). Les six dernières colonnes correspondent au classement final et au nombre de points final (c'est donc une répétition qui ne correspond à aucun match). Enfin le milieu de saison est répété et correspond à un mois de trêve en hiver,

Au minimum, vous implémenterez le chargement des données avec un `ifstream` et un `getline(file, value, c);` qui lit le fichier `file` et met dans la string `value` le résultat de la lecture jusqu'au prochain

```
Man City, 5, 0, ??,-1,-1,??,5, 3, Arsenal,0, 2, Man City, 0, 6, Man City,6, 1, Huddersfield, 4, 7, Wolves,1, 1, Man City, 3,
10, Man City,2, 1, Newcastle, 2, 13, Man City,3, 0, Fulham, 1, 10, Cardiff,0, 5, Man City,0, 19, Man City,2, 0, Brighton, 0,
20, Liverpool,0, 8, Man City,0, 23, Man City,5, 0, Burnley, 0, 20, Tottenham,0, 1, Man City,0, 29, Man City,6, 1,
Southampton, 0, 32, Man City,3, 1, Man United, 0, 35, West Ham,0, 4, Man City,0, 30, Man City,3, 1, Bournemouth, 0, 41,
Watford,1, 2, Man City, 1, 41, Chelsea,2, 0, Man City, 1, 44, Man City,3, 1, Everton, 1, 44, Man City,2, 3, Crystal Palace, 2,
44, Leicester,2, 1, Man City, 2, 44, ??,-1,-1,??,1, 47, Southampton,1, 3, Man City, 1, 50, Man City,2, 1, Liverpool, 1, 53,
Man City,3, 0, Wolves, 1, 50, Huddersfield,0, 3, Man City, 1, 50, Newcastle,2, 1, Man City, 1, 50, Man City,3, 1, Arsenal, 1,
62, Man City,6, 0, Chelsea, 1, 65, Everton,0, 2, Man City, 1, 68, Man City,1, 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, 86, Crystal Palace,1, 5, Man City, 0, 89, Man City,1, 0, Tottenham, 0, 92, Burnley,0, 1, Man City, 0, 95, Man City,1, 0,
Leicester, 0, 98, Brighton,1, 4, Man City, 0, 98, ??,-1,-1,??,
Liverpool, 0, 0, ??,-1,-1,??,0, 3, Liverpool,1, 0, West Ham, 1, 5, Crystal Palace,0, 2, Liverpool, 0, 9, Liverpool,1, 0,
Brighton, 0, 12, Leicester,1, 2, Liverpool, 1, 15, Tottenham, 1, 2, Liverpool,0, 10, Liverpool,3, 0, Southampton, 1, 19,
Chelsea,1, 1, Liverpool, 2, 20, Liverpool,0, 0, Man City, 1, 23, Huddersfield,0, 1, Liverpool, 1, 26, Liverpool,4, 1, Cardiff,
2, 27, Arsenal,1, 1, Liverpool, 1, 30, Liverpool,2, 0, Fulham, 1, 35, Watford,0, 3, Liverpool, 1, 36, Liverpool,1, 0, Everton,
1, 39, 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,4, 0, Newcastle, 0, 51, ??,-1,-1,??,0, 54, Liverpool,5, 1, Arsenal, 0, 54, Man City,2, 1,
Liverpool, 0, 57, Brighton,0, 1, Liverpool,0, 60, Liverpool,4, 3, Crystal Palace,0, 61, Liverpool,1, 1, Leicester, 0, 62,
West Ham,1, 1, Liverpool, 0, 65, Liverpool,3, 0, Bournemouth, 0, 66, Man United, 0, 69, Liverpool,5, 0, Liverpool,5, 0,
Watford, 1, 70, Everton,0, 0, Liverpool, 1, 73, Liverpool,4, 2, Burnley, 1, 76, Fulham,1, 2, Liverpool, 1, 79, Liverpool,2, 1,
Tottenham, 1, 82, Southampton, 1, 2, Liverpool, 1, 85, Liverpool,2, 0, Chelsea, 1, 88, Cardiff,0, 2, Liverpool, 1, 91,
Liverpool,5, 0, Huddersfield, 1, 94, Newcastle,2, 3, Liverpool, 1, 97, Liverpool,2, 0, Wolves, 1, 97, ??,-1,-1,??,
Chelsea, 1, 0, ??,-1,-1,??,1, 3, Huddersfield,0, 3, Chelsea, 2, 6, Chelsea,3, 2, Arsenal, 2, 9, Newcastle,1, 2, Chelsea, 1,
12, Chelsea,2, 0, Bournemouth, 0, 15, Chelsea, 4, 1, Cardiff, 2, 16, West Ham,0, 0, Chelsea, 2, 17, Chelsea,1, 1, Liverpool, 1,
20, Southampton,0, 3, Chelsea, 2, 21, Chelsea,2, 2, Man United, 2, 24, Burnley,0, 4, Chelsea, 1, 27, Chelsea,3, 1, Crystal
Palace, 2, 28, Chelsea,0, 0, Everton, 3, 28, Tottenham,1, 1, Chelsea, 3, 31, Chelsea,2, 0, Fulham, 3, 31, Wolves,2, 1,
Chelsea, 3, 34, Chelsea,2, 0, Man City, 3, 37, Brighton,1, 2, Chelsea, 3, 37, Chelsea,0, 1, Leicester, 3, 40, Watford,1, 2,
Chelsea, 3, 40, ??,-1,-1,??,3, 43, Crystal Palace,0, 1, Chelsea, 3, 44, Chelsea,0, 0, Southampton, 3, 47, Chelsea,2, 1,
Newcastle, 3, 47, Arsenal,2, 0, Chelsea, 4, 47, Bournemouth,4, 0, Chelsea, 3, 50, Chelsea,5, 0, Huddersfield, 5, 50, Man
City,6, 0, Chelsea, 3, 53, Chelsea,3, 0, 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,2, 0, Chelsea, 3, 67, Chelsea,2, 2, Burnley, 3, 68, Man United,1, 1, Chelsea, 2, 71, Chelsea,3, 0, Watford, 2, 72,
Leicester,0, 0, Chelsea, 2, 72, ??,-1,-1,??,
Tottenham, 7, 0, ??,-1,-1,??,7, 3, Newcastle,1, 2, Tottenham, 4, 6, Tottenham,3, 1, Fulham, 1, 9, Man United,0, 3, Tottenham,
4, 9, Watford,2, 1, Tottenham, 5, 9, Tottenham,1, 2, Liverpool, 4, 12, Brighton,1, 2, Tottenham, 3, 15, Huddersfield,0, 2,
Tottenham, 4, 10, Tottenham,1, 0, Cardiff, 4, 21, West Ham,0, 1, Tottenham, 4, 21, Tottenham,0, 1, Man City, 3, 24, Wolves,2,
3, Tottenham, 3, 27, Crystal Palace,0, 1, Tottenham, 3, 30, Tottenham,3, 1, Chelsea, 4, 30, Arsenal,4, 2, Tottenham, 2, 33,
Tottenham,3, 1, Southampton, 2, 36, Leicester,0, 2, Tottenham, 2, 39, Tottenham,1, 0, Burnley, 2, 42, Everton,2, 6, Tottenham,
1, 45, Tottenham, 0, 0, Bournemouth, 1, 45, ??,-1,-1,??,2, 45, Tottenham,1, 3, Wolves, 2, 46, Cardiff,0, 2, Tottenham, 2, 48,
Tottenham,0, 1, Man United, 2, 51, Fulham,1, 2, Tottenham, 2, 54, Tottenham,2, 1, Watford, 2, 57, Tottenham,1, 0, Newcastle,
```

caractère **c**. Les fonctions **stoi(...)** convertissent une **string** en **int**. Vous devez implémenter la fonction **loadData** suivante :

```
void loadData(string fn, vector<int> &pts, vector<int> &ranks);
```

qui lit le fichier **fn** et remplit les vecteurs **pts** et **ranks** avec les colonnes impaires et paires.

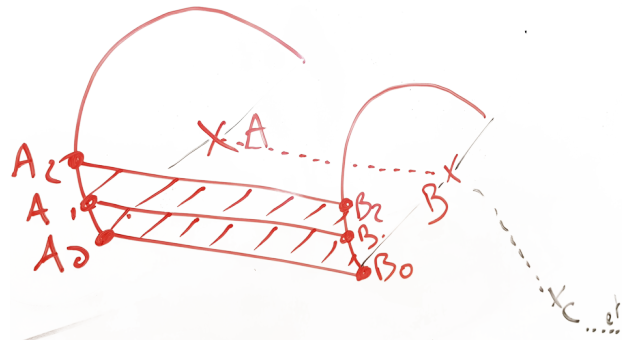
Pour aller un peu plus loin, vous pourrez charger les textures (1 par équipe) avec la librairie openCV. Vous stockerez les IDs dans un tableau dont les indices correspondront au numéro de ligne du fichier csv, vous chargerez aussi les informations du match qui a conduit à ce classement et ce nombre de point.

Pour aller plus loin, vous pourrez charger dans le fichier détaillé (englishR.csv) une ligne pour chacun des 380 matchs, vous trouverez la date, la localisation du match, la côte des paris sportifs, etc.). Voir la description des données ici qui correspond aux titres de colonne:

<https://www.football-data.co.uk/notes.txt>

Modèles 3D

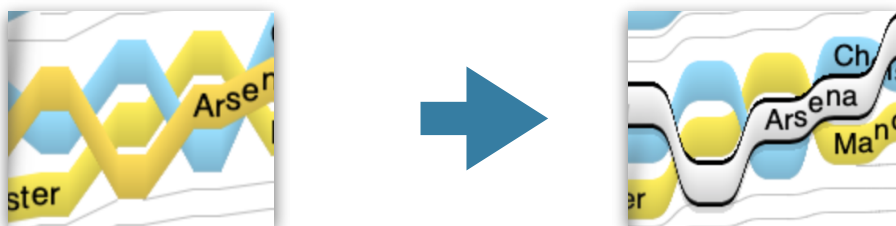
Au minimum, vous implémenterez le chargement de 20 VAO. Chaque VAO contient les points d'une courbe sous forme de demi-cylindres successifs. Entre 2 points successifs, vous calculerez les points A_i et B_i et formerez des TRIANGLES ou des QUADS avec le bon vecteur normal (si vous réfléchissez bien pas besoin de produit vectoriel pour le calculer !)



Pour chaque équipe un gapchart affiche une courbe d'une épaisseur $(height/2)/20 =$ la moitié de la hauteur, divisée par 20 équipes. La position en y d'une courbe à un instant t est un mélange du complémentaire du classement (19-le classement) et du nombre de point. Pour être mélangées ces deux valeurs doivent être normalisées (le complémentaire du classement est divisé par 19 et le nombre de point est divisé par le nombre maximum de point = 98 dans le fichier exemple). Le mélange doit enfin être re-multiplié par height. Un exemple en ligne de gapchart en 2D peut être testé ici :

Entre deux temps successifs un gapchart affiche d’abord un morceau de demi-cylindre horizontal, puis un morceau de demi-cylindre “penché” qui relie au début du prochain demi-cylindre horizontal (dont la position est calculée par $t+1$, le classement au $t+1$ et le nombre de points à $t+1$).

Pour aller un peu plus loin, vous pourrez calculer des étapes intermédiaires pour la partie penchée qui donnent un effet de S (au lieu d'une diagonale droite)



LICENCE D'INFORMATIQUE : PROJET WEB

Pour aller plus loin, vous pourrez afficher 38 arcs de cercle qui relient les opposants d'une équipe. La hauteur de l'arc de cercle entre les deux points pourra être relative à la différence de classement ou bien à une autre donnée (ie la cote des paris sportifs,, le nombre de buts, etc.)

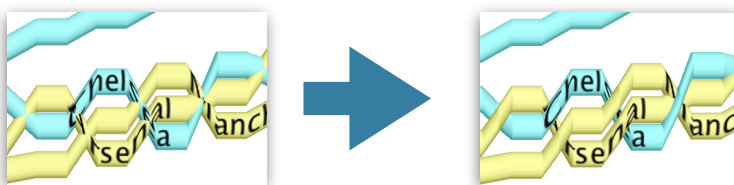
```
int loadModel(string team, vector<int> &pts, vector<int> &ranks, GLuint VertexArrayID);
```

cette fonction retourne le nombre de « vertices »/sommets créés.

Couleurs et ombrages

Au minimum, les quatre premières courbes seront bleues, les 3 suivantes jaunes et les 4 dernières seront rouges. Vous implémenterez un ombrage (assombrissement) qui donne du relief aux courbes ET qui fasse en sorte que se soit toujours la courbe qui monte qui passe "devant" la courbe qui descend quand 2 courbes se croisent. Pour commencer vous changerez la position des sommets "penchés" en fonction du gain/perte de rang.

Exemple : remarquez l'intersection de la courbe bleue et des courbes jaunes. A gauche les tuyaux se chevauchent, à droite c'est le tuyau qui monte qui passe devant.



Pour aller un peu plus loin, vous appliquerez les textures sur les demi-cylindres pour faire apparaître les noms des équipes (mais si la texture est transparente pour un pixel donné on utilisera la couleur "assombrie"). Vous tenterez également de faire un ombrage différent selon le résultat du match (+3pts = match gagné, +1pt = match nul, +0pt= match perdu)

Pour aller un plus loin, Pour réaliser l'effet devant/derriere, vous transmettez l'information de gain/perte de rang au shader et modifierez la profondeur en conséquence. Le shading des cylindres ne sera donc pas modifié par un déplacement des sommets !

Interaction

Au minimum, vous permettrez le déplacement de la camera en 3D autour du modèle (avec des touches du clavier) et vous permettrez de sélectionner l'une des 20 équipes (ie avec 2 touches) en changeant son rendu (ie en augmentant son z pour le faire passer devant).

Pour aller plus loin, vous rendrez les variantes de l'algorithme accessibles par des touches (une seule touche pour activer/désactiver un rendu, plusieurs touches pour des variantes, 2 touches pour diminuer/augmenter un paramètre). Vous pourrez aussi utiliser la souris pour les déplacements de caméra.

Rendu final

Chaque étudiant/binôme devra remettre à l'enseignant un rapport (entre 3 et 4 pages avec les images) organisé afin de décrire comment vous avez utilisé les technologies demandées, un code C++ fonctionnel et commenté (dans une archive zip à votre nom). En outre chaque étudiant/binôme devra faire une démonstration devant les enseignants lors de la dernière séance afin de défendre son approche, ses choix et montrer les possibilités des interactions.

Bonne chance !

