RAPPORT – MINI-PROJET – 2021-2022

HIDALGO SARAH – FROUTÉ MARIE

1 . Spécifications

L'objectif était de mettre en place un trombinoscope cliquable pour permettre aux professeurs de Télécom Saint-Étienne de faire l'appel. L'application commence par une interface console où il est demandé à l'utilisateur de saisir la filière (pour permettre une utilisation plus générale) ainsi que le groupe d'étudiants dont il souhaite faire l'appel. Pour la saisie, l'utilisateur est guidé par un exemple de la syntaxe requise affiché au-dessus de la question.

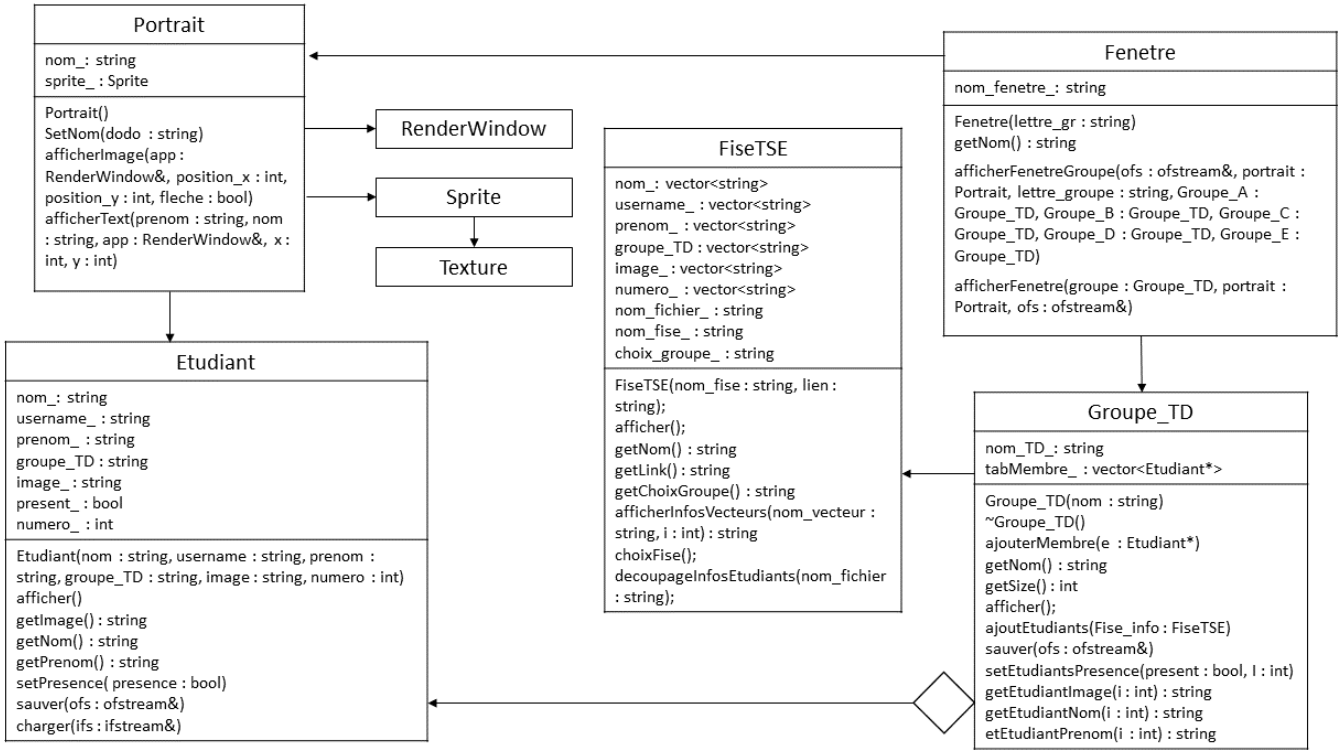
Ensuite, une fenêtre s'affiche avec le trombinoscope du groupe choisi. Tous les étudiants ne passent pas dans une seule fenêtre, donc on a mis en place des flèches de navigation sur les côtés latéraux de l'écran pour permettre de faire défiler les autres étudiants. Lors d'un clic de souris gauche sur ces flèches, on change de page. Une page contient 12 élèves au maximum.

Les élèves se font attribuer le statut absent par défaut et si l'utilisateur clic sur leur image, leur statut devient présent. Le statut est indiqué via une image correspondant à une lettre P pour présent et A pour absent, en haut à gauche de la vignette de l'élève. De manière générale, l'appui sur une l'image change l'état (si l'état était absent, il devient présent et vice versa).

À la fermeture de la fenêtre d'affichage, un fichier .txt nommé Fiche d'appel est créé à la racine de notre projet. Celui-ci contient tout d'abord le nom du groupe et son effectif, et ensuite la liste des étudiants qui le compose ainsi que leurs attributs et leur statut d'absent ou présent. Chaque étudiant est séparé par une ligne de pointillés.

2. Diagramme des classes

Notre projet contient 5 classes : Etudiant, Groupe_TD, FiseTSE, Portrait, Fenetre.



FiseTSE permet de gérer le fichier csv, comme son découpage, et de gérer l'échange avec l'utilisateur.

```
#pragma once

#include <string>
#include <fstream>
#include <iostream>
#include <vector>

using namespace std;

// Créée par Sarah HDG
class FiseTSE
{
    string nom_fichier_, nom_fise_, choix_groupe_; // nom du fichier.csv, nom de la fise, nom du groupe ("GrA")
    vector<string> numero_, username_, nom_, prenom_, groupe_TD_, image_; // vecteur numero_ contient tout les numeros du fichier,
    // même principe pour les autres vecteurs
public:
    FiseTSE(string nom_fise = " ", string lien = " "); //constructeur par defaut de la classe FIseTSE
    void afficher(); //affichage des parametres de la classe sur la console
    const string& getNom() const { return nom_fise_; };
    const string& getLink() const { return nom_fichier_; };
    const string& getChoixGroupe() const { return choix_groupe_; };
    string afficherInfosVecteurs(string nom_vecteur, int i); // return nom_vecteur[i] (ex : prenom_[3] le prenom de l'etudiant numéro 4)
    void choixFise(); //sert a rentrer son choix de FISE et de groupe de TD sur la console
    void decoupageInfosEtudiants(string nom_fichier); //tri du fichier .csv pour remplir les vecteurs
};

```

Groupe_TD va permettre de rassembler les étudiants d'un même groupe dans une seule entité.

```cpp
#pragma once

#include <string>
#include <vector>
#include <fstream>

#include "Etudiants.h"
#include "FiseTSE.h"

using namespace std;

// Créée par Marie FRT et Sarah HDG
class Groupe_TD : public FiseTSE
{
    string nom_TD_; // lettre du groupe de TD
    vector<Etudiant*> tabMembre_; //vecteur composer de pointeur vers etudiants

public:
    Groupe_TD(const string& nom = "");// constructeur par defaut de la classe Groupe_TD
    ~Groupe_TD();
    const string& getNom() const { return nom_TD_; }
    int getSize() const { return tabMembre_.size(); }
    void afficher();
    void ajoutEtudiants(FiseTSE Fise_info); // ajout des etudiants dans le groupe de TD
    void sauver(ofstream& ofs) const; // sauvegarde des eleves du groupe du TD dans un fichier
    void setEtudiantsPresence(bool present, int i) { tabMembre_[i]->setPresence(present); } // modifie le bool indiquant la presence
    // de l'élève d'indice i en bool present
    string getEtudiantImage(int i) { return tabMembre_[i]->getImage(); } //recup de l'image de l'étudiant d'indice i
    string getEtudiantNom(int i) { return tabMembre_[i]->getNom(); } //recup du nom de l'étudiant d'indice i
    string getEtudiantPrenom(int i) { return tabMembre_[i]->getPrenom(); } //recup du prenom de l'étudiant d'indice i
};
```

Etudiant est la classe de base qui permet de stocker les données des étudiants, fournies par le fichier csv.

```cpp
#pragma once

#include <string>
#include <fstream>
#include <iostream>

using namespace std;

// Créée par Sarah HDG
class Etudiant
{
    string nom_, username_, prenom_, groupe_TD_, image_;
    bool present_; // bool qui indique la présence de l'élève
    int numero_; //num d'identification de l'etudiant dans le fichier
public:
    Etudiant(string nom = " ", string username = " ", string prenom = " ", string groupe_TD = " ", string image = " ", int numero=0, bool present=false);
    //constructeur de la classe étudiant
    void afficher(); //afficher les paramètres de l'étudiants sur la console
    const string& getImage()const { return image_; };
    const string getNom() { return nom_; };
    void setPresence(bool presence) { present_ = presence; } // modifie le booléen present_
    const string getPrenom() { return prenom_; };
    void sauver(ofstream& ofs) const; // sauvegarder les parametres de l'etudiants dans un fichier
    //void charger(ifstream& ifs);
};

```

Portrait va permettre de gérer l'association des fichiers images à un Sprite, en utilisant le nom de l'image de chaque étudiant (information contenue dans le fichier csv) pour aller chercher l'image en elle-même avec le chemin relatif désormais connu.

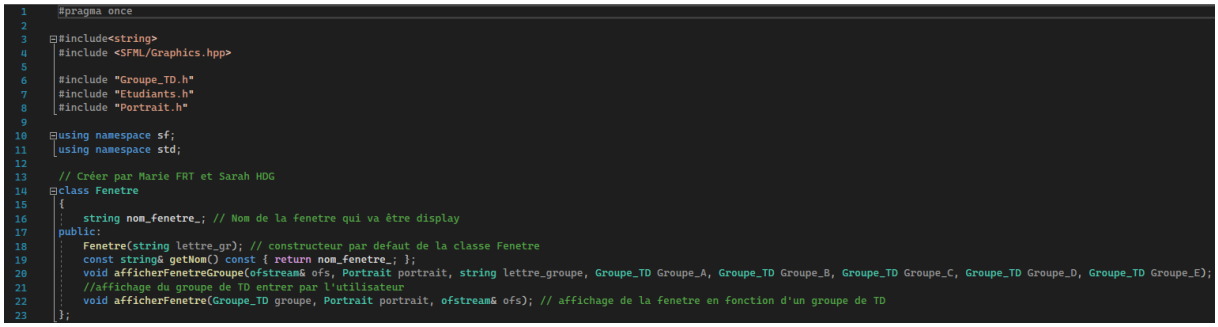
```cpp
#pragma once

#include<string>
#include <SFML/Graphics.hpp>
#include "Groupe_TD.h"
#include "Etudiants.h"

using namespace sf;
using namespace std;

// Créée par Marie FRT, 1 méthode ajoutée par Sarah
class Portrait
{
    string nom_; // nom de l'image a ajouter a la texture .png
    Sprite sprite_;
public:
    Portrait();
    void SetNom(string dodo) { nom_ = dodo; }; // modifie le nom de l'image mettre dans la texture
    void afficherImage(RenderWindow&, int position_x, int position_y, bool fleche); // fonction qui draw une image
    void afficherText(string prenom, string nom, RenderWindow&, int x, int y); // fonction qui draw un texte
};
```

Fenetre est la classe qui gère l'affichage des fenêtres du trombinoscope ainsi que tous les évènements liés à l'utilisation du programme (comme le clic sur une image ou la navigation avec les flêches).



```h
#pragma once

#include<string>
#include <SFML/Graphics.hpp>

#include "Groupe_TD.h"
#include "Etudiants.h"
#include "Portrait.h"

using namespace sf;
using namespace std;

// Créer par Marie FRT et Sarah HDG
class Fenetre
{
    string nom_fenetre_; // Nom de la fenetre qui va être display
public:
    Fenetre(string lettre_gr); // constructeur par defaut de la classe Fenetre
    const string& getNom() const { return nom_fenetre_; };
    void afficherFenetreGroupe(ofstream& ofs, Portrait portrait, string lettre_groupe, Groupe_TD Groupe_A, Groupe_TD Groupe_B, Groupe_TD Groupe_C, Groupe_TD Groupe_D, Groupe_TD Groupe_E);
    //affichage du groupe de TD entrer par l'utilisateur
    void afficherFenetre(Groupe_TD groupe, Portrait portrait, ofstream& ofs); // affichage de la fenetre en fonction d'un groupe de TD
};
```