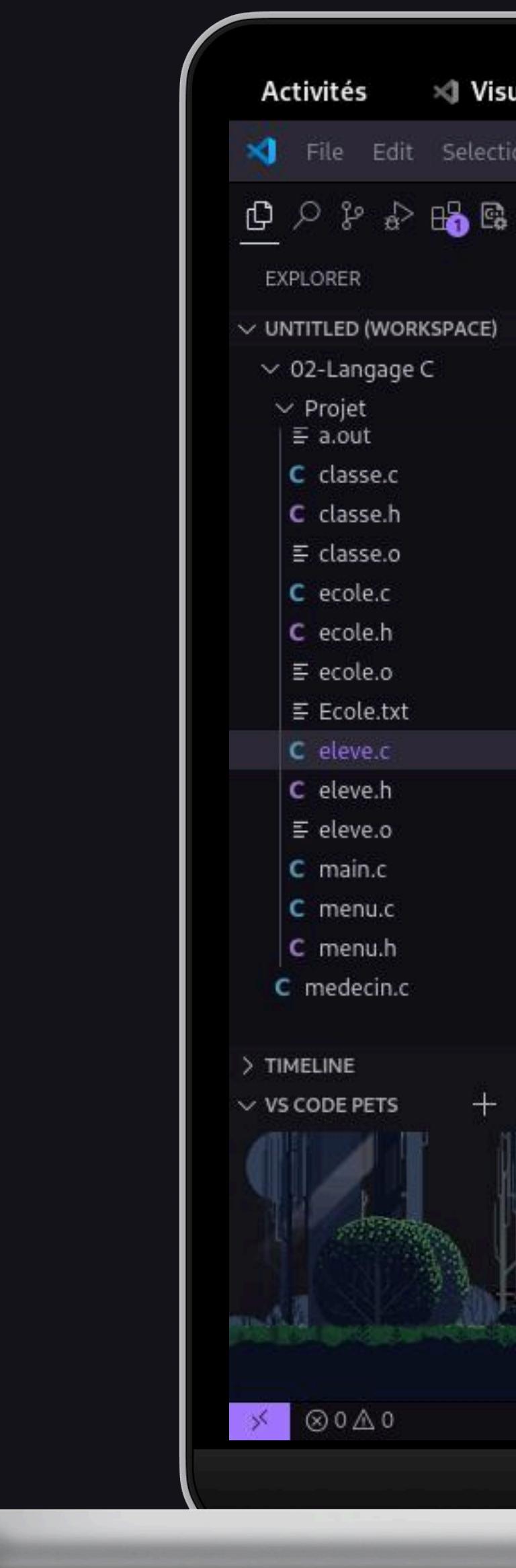


# ECOLE.EXE



- 01
- 02 Elias BOUNASMIR
- 03 Jean-Sebastien SALLES
- 04 Tristan REGNIER
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12

# SOMMAIRE:



# SOMMAIRE:

- PRÉSENTATION DE L'ÉQUIPE
  - OBJECTIF DU PROGRAMME
  - ORGANISATION DU TRAVAIL
  - REPARTITION DES TACHES
  - ORGANISATION DES FICHIERS
  - STRUCTURES
  - DÉMONSTRATION DU PROGRAMME
  - POINTS PARTICULIÉS
  - AXES D'AMÉLIORATION

Activités Visual Studio Code 12 juin 15:09

File Edit Selection View Go Run ... ← → ⌂ Untitled (Workspace)

EXPLORER ...

UNTITLED (WORKSPACE)

02-Langage C > Projet > eleve.c > SaisirEleve(eleve\_t \*)

classe.c classe.h ecole.c eleve.c eleve.h main.c Projet

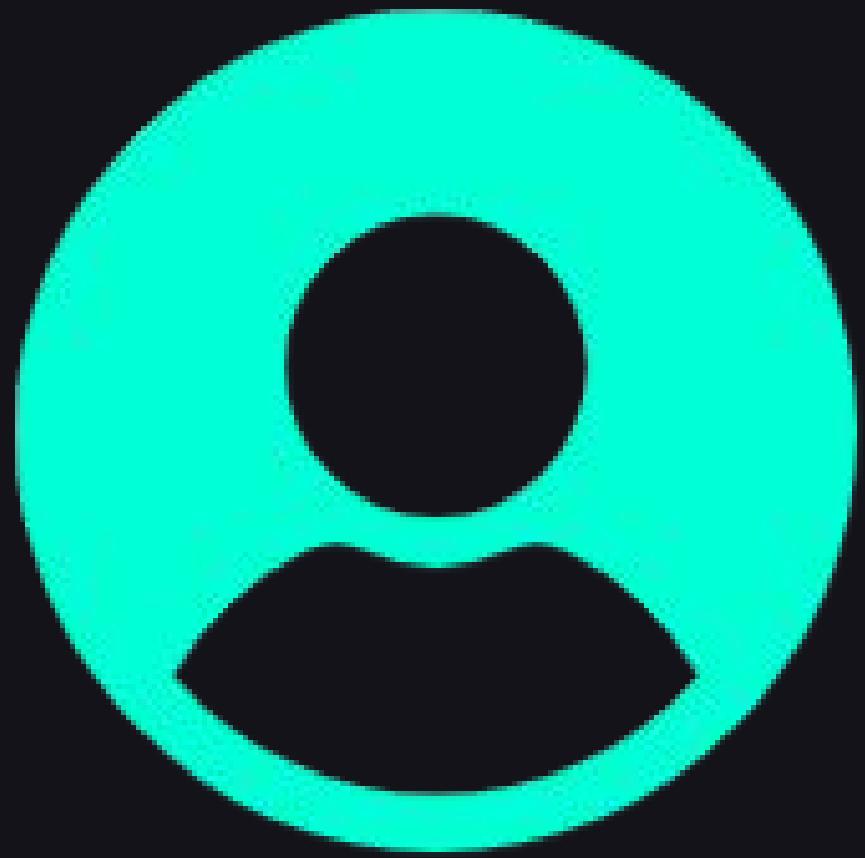
```
47 void SaisirEleve(eleve_t *A){  
48     do {  
49         } while (!(temp == 'H') || (temp == 'F')); // Tant que l'utilisateur n'a pas renseigné H ou F  
50         switch (temp){ // En fonction de la réponse on assigne la bonne valeur à A.sex  
51             case 'H':  
52                 A->sex = 1; // Passer le sexe à Homme  
53                 break;  
54             default:  
55                 A->sex = 0; // Passer le sexe à Femme  
56                 break;}  
57         // Saisie date de naissance de l'élève -----|  
58         printf("Quel est la date de naissance de %s %s ? (dd/mm/yyyy)\n", A->prenom,  
59             scanf("%2d/%2d/%4d", &A->dateNaissance.tm_mday, &A->dateNaissance.tm_mon, &A->dateNaissance.tm_year);  
60             A->dateNaissance.tm_mon -= 1; // Les mois commencent à 0  
61             A->dateNaissance.tm_year -= 1900; // Les années commencent à 1900  
62         // Appeler la fonction AffecterEleveClasse() pour définir la classe de l'élève  
63         AffecterEleveClasse(A);  
64     }  
65     // ----- IMPRIMER ELEVE -----  
66     void AfficherEleve(const eleve_t *A){...}  
67     void ModifierEleve(eleve_t *A){...}  
68     // ----- IMPRIMER ELEVE -----  
69     void ImprimerEleve(const eleve_t *A, char *nomFichier){...}
```

TIMELINE

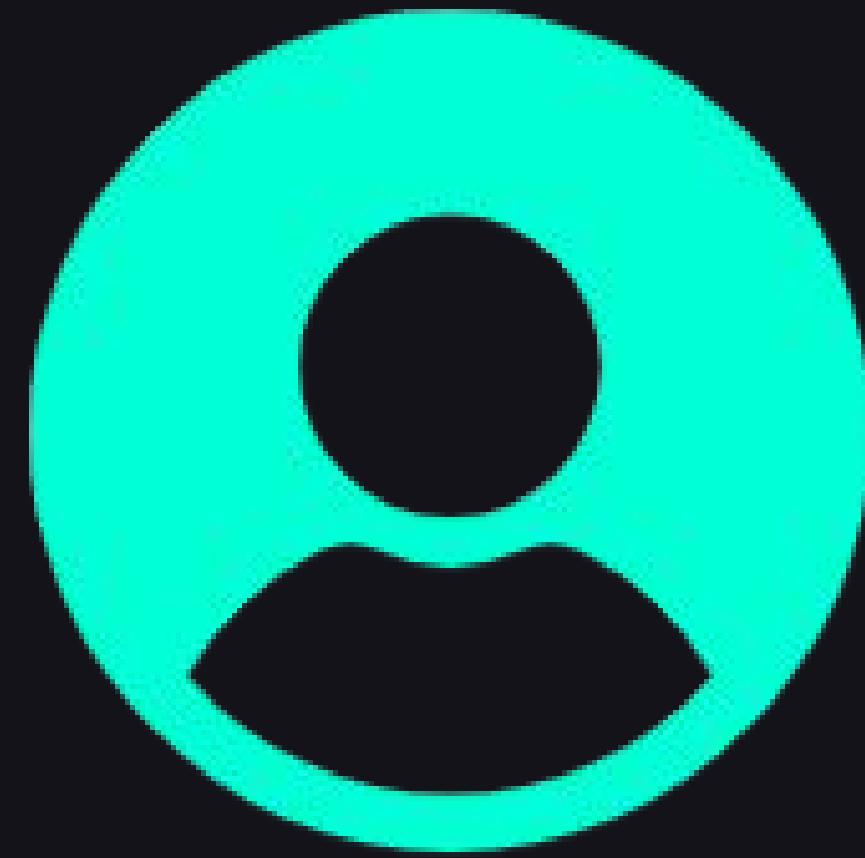
VS CODE PETS + o ⚡

0 △ 0 Ln 71, Col 67 Sp

# Présentation De L'équipe



ELIAS BOUNASMIR  
aka: coffee-reponsable



JEAN-SEBASTIEN SALLES  
aka: Bug-Destroyer



TRISTAN REGNIER  
aka: weird-idea-manager

# Objectif Du Programme Et Fonction Principales



## SÉNARIO :

Création d'un programme pour **une école**. Le programme est destiné au directeur de l'école. Il doit lui permettre d'**accéder aux informations de son école** :

- Nombre de **classes**
- Nombre de classes par **Niveaux**
- Nombre **d'élèves** par classe
- Nom des **professeurs**
- Rechercher un élève et les **informations** qui lui sont associés
- **Modifier** les informations d'un élève

# Organisation Du Travail

**LDNR PROJECT – FigJam**

[figma.com/board/LXFrqmW0fiGY5x8rsTGogw/LDNR-PROJECT?node-id=0-1&p=f&t=0kMms4qN3Ge3eeXd-0](https://figma.com/board/LXFrqmW0fiGY5x8rsTGogw/LDNR-PROJECT?node-id=0-1&p=f&t=0kMms4qN3Ge3eeXd-0)

**Chemin saisir élève**

**Chemin modifier élève**

**BASE ECOLE**

- Trouver tous les noms de classe présent dans l'école
- Int nbClasse[] : Le tableau qui récupère le nombre d'élèves total par section puis nombre total de classes

**Struct tabClasse\_t [NIVEAU CLASSE(6)][NUMERO CLASSE(3 classe)]**

- char nom[30] : Ce sera le nom de la classe
- int nbElles : Ce sera le nombre d'élèves dans une classe, attention max 25 personnes par classe
- struct tabEleve[] : La structure personne qui désignera le prof dans les classes
- int regars, nbElles;

**struct tabEleve[]**

- char nom[30];
- char prenom[30];
- int sexe; (1 = M | 0 = F)
- struct tabEleve \*next;
- int nbElles; -> Information sur le nombre de la classe

**FONCTIONS :**

- void SaisirEleve(tabEleve\_t \*t, int nbrClasse) : Ajouter à la structure personne toutes les informations
- void AfficherEleve(const tabEleve\_t \*t) : Afficher les informations sur un élève (Nom)
- void ModifierEleve(tabEleve\_t \*t) : Modifie les informations d'une personne.
- void AffecterEleveClasse(tabEleve\_t \*t) : Qui affecte une personne à une classe par rapport à l'age. Faire une vérification si la date de naissance est dans le scope des classes (2015 < DOB < 2018)

**FONCTIONS :**

- void AfficherTabEleveClasse\_t \*t) : Afficher toutes les personnes dans une classe.
- void RecupererEleveClasse(tabEleve\_t \*\*t, const char \*nom, const char \*prenom, int \*nbEllesPersonne) : Crée une nouvelle liste appeler la fonction AffecterEleve()
- void SaisirProf(tabClasse\_t \*t) : Qui demande en main les prenoms d'un prof dans une classe donnée
- void AfficherClasse(tabClasse\_t \*t) : Affiche le nom de la classe et le nom, prénom du prof

**nbClasse : tableau qui compte le nombre d'élèves total par niveau (ex CM1 ou CM2 il déclenche à chaque tour de boucle while).**

- void AfficherClasse(tabClasse\_t \*t) : Boucle while qui, autant de fois que l'on souhaite va boucler sur tabClasse.
- SaisirEleve() va automatiquement AffecterEleve() et AffecterEleve() va automatiquement incrementer le tableau nbClasse en fonction de la DOB.
- Copier l'élève temporaire dans tabClasse[1][1] en fonction de sa DOB.

**tabClasse : tableau qui compte le nombre d'élèves total par niveau (ex CM1 ou CM2 il déclenche à chaque tour de boucle while).**

- void AfficherClasse(tabClasse\_t \*t) : Afficher l'ensemble des classes et des personnes dans les classes.
- void RecupererEleveClasse() const char \*nom, const char \*prenom, int \*nbClasse, int \*nbNiveauClasse, int \*nbEllesPersonnes) : Fonction qui déclenche l'insertion des éléments en appuyant sur la fonction RecupererEleveClasse()
- void ExtraireClasse(tabClasse\_t \*tabClasse, int nbClasse) : Extraire les élèves de façon déplaçable dans les classes.
- void RepartirClasse(tabClasse\_t \*tabClasse, int nbClasse) : Repartir les élèves de façon déplaçable dans les classes.
- void RepartirClasse(tabClasse\_t \*t) :

**SUPER PROCEDURE**

```

1 //INITIALISATION DES VARIABLES
2 int statut = 0;
3 char valide = 1;
4
5 //INITIALISATION DES TABLEAUX
6 print("-----SUPER PROCEDURE-----");
7 print("-----#1-----");
8 print("-----#2-----");
9 print("-----#3-----");
10 print("-----#4-----");
11 print("-----#5-----");
12
13 //DATA init
14
15 //INTERROGATION DANS LA BANQUE DE DONNEES
16 print("Saisissez une option de menu (1-5)");
17 scanf("%d", &choix);
18
19 if (choix == 1)
20 {
21     print("Veuillez choisir l'option suivante");
22     valide = 0;
23 }
24 else if (choix == 2)
25 {
26     print("Veuillez choisir l'option suivante");
27     valide = 0;
28 }
29 else if (choix == 3)
30 {
31     print("Veuillez choisir l'option suivante");
32     valide = 0;
33 }
34 else if (choix == 4)
35 {
36     print("Veuillez choisir l'option suivante");
37     valide = 0;
38 }
39 else if (choix == 5)
40 {
41     print("Saisissez une option de menu (1-5)");
42     valide = 1;
43 }

```

05/11

# Organisation Du Travail

The screenshot shows a FigJam project titled "LDNR PROJECT – FigJam". The main content is a diagram titled "BASE ECOLE" with a yellow border. It contains several sections:

- maVariableEstSuper**: A note about variable naming.
- Fonction commencent par une majuscule :** A note about function naming.
- TrierLaListe()**: A note about sorting functions.
- maVariableEstSuper**: A note about variable naming.
- Chemin saisir eleve**
- Chemin modifier eleve**
- STRUCTURE :** Descriptions of **struct tabClasse\_t** and **struct tabEleve[ ]**.
- FONCTIONS :** Descriptions of various functions like **SaisirEleve**, **AfficherEleve**, **ModifierEleve**, and **AffecterEleveClasse**.
- STRUCTURE :** Description of **struct prof[ ]**.
- FONCTIONS :** Description of **EstInscrit**.
- Tableau classe**: A table showing classes and their details.
- CODE C**: A snippet of C code for a menu system.

Annotations include a thumbs-up icon on the left, a red fire icon above the class structure, a green thumbs-down icon above the professor structure, and a blue arrow pointing from the student structure to the professor structure. There are also purple arrows connecting the student and professor structures to the class table.

```
1 //Déclaration des variables
2 int choix = 0;
3 char valide = 'n'
4
5 //Affichage menu
6 printf("----- SUPER PROGRAMME -----")
7 printf("-- a1")
8 printf("-- a2")
9 printf("-- a3")
10 printf("-- a4")
11 printf("-----")
12
13 //Choix menu
14 do
15 {
16     printf("Choisissez une option du menu : \n");
17     scanf(" %d", &choix);
18
19     if ((choix == 1))
20     {
21         printf("Vous avez choisi l'option %d\n");
22         //---code---
23         valide = 'y';
24     }
25     else if ((choix == 2))
26     {
27         printf("Vous avez choisi l'option %d\n");
28         //---code---
29         valide = 'y';
30     }
31
32     else if ((choix == 3))
33     {
34         printf("Vous avez choisi l'option %d\n");
35         //---code---
36         valide = 'y';
37     }
38
39     else{
40         printf("Saisissez une option valide : \n");
41     }
42 } while (valide != 'y');
```

# Répartition Des Taches

Elias Bounasmir :

- structure ecole\_t
- Menu de navigation
- fonction() LireFichier

Jean Sébastien Salles :

- strucutre classe
- makeFile
- fonction() classe

Tristan Regnier :

- structure eleve\_t
- fonction() relatives a eleve\_t
- Diaporama présentation

Travail en commun :

- fonction() relative a la structure ecole\_t
- Débugage général



# Organisation Des Fichiers

main.c - X

menu.c - menu.h - X

fichier.c - fichier.h - X

ecole.c - ecole.h - X

classe.c - classe.h - X

eleve.c - eleve.h - X



# Structures

Fichier : ecole.h

```

18 //structure école
19 typedef struct ecole
20 {
21     char nom[MAXCHAR];
22     directeur_t directeur;
23     classe_t tabClasse[NIVEAUCLASSE][NUMEROCLASSE];
24     int nbClasse[NIVEAUCLASSE];
25 }

```

Fichier : classe.h

```

17 typedef struct classe
18 {
19     char nom[MAXCHAR];
20     int nbrEleve;
21     prof_t prof;
22     int nbgars;
23     int nbfille;
24     eleve_t tabPersonne[MAXELEVE];
25 }

```

Fichier : ecole.h

```

10 //structure directeur
11 typedef struct directeur
12 {
13     char nom[MAXCHAR];
14     char prenom[MAXCHAR];
15 }
16

```

Fichier : classe.h

```

11 typedef struct prof
12 {
13     char nom[MAXCHAR];
14     char prenom[MAXCHAR];
15 }

```

Fichier : eleve.h

```

15 typedef struct eleve {
16     char nom[MAXCHAR];
17     char prenom[MAXCHAR];
18     bool sexe;
19     struct tm dateNaissance;
20     int numClasse;

```

# DEMONSTRATION DU PROGRAMME



# Difficultés Rencontrés

- Choix d'une structure trop complexe : tableau à plusieurs dimensions
- Gestion du versioning (GitHub aurait été sympa ... )
- Gestion du temps, il aurait été préférable de faire + de conception avant de démarrer





## Axes D'améliorations

*I*

Ajouter un fonction de tri des élèves par ordre alphabetique

*II*

Ajouter un fonction de répartition équitable en genre dans les classes

*III*

Terminer la fonction qui permet de modifier la date de naissance d'un élève et par consequent reaffecter et re-répartir les classes.

*IV*

Terminer la fonction qui permet de modifier toutes les informations d'une école.

*V*

Choisir le format d'impression des fichiers : impression générale de l'école ou impression par classe.

*VI*

Ajouter une fonction qui permet de gerer les heures de colles et les punissements et les redoublements pour les élèves ...



```
Printf(" Merci Pour  
Votre Attention ");  
// AVEZ VOUS DES QUESTIONS ?
```

*ELIAS  
BOUNASMIR*

*JEAN-SEBASTIEN  
SALLES*

*TRISTAN  
REGNIER*