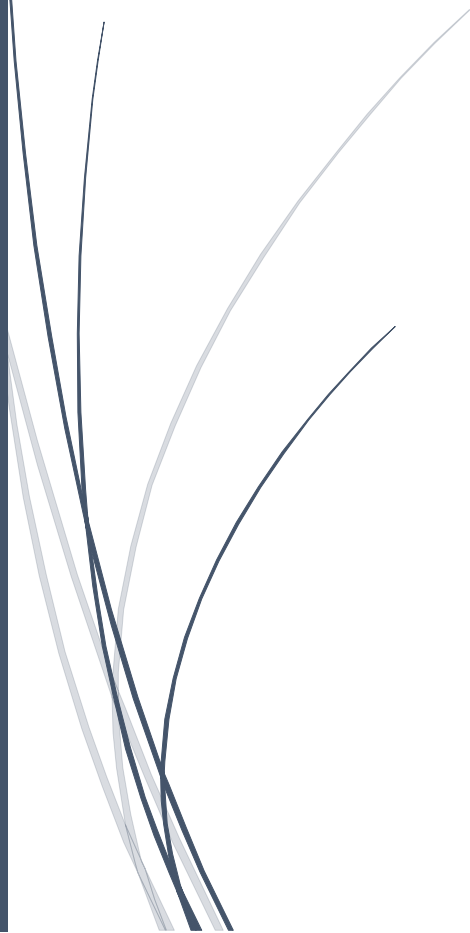


A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

20/12/2021

# Rapport projet : Cinéma

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Kyllian ALLAND - Jean-Sébastien CAMBIER  
2BIA2B2

## Table des matières

Explication de la notion de liste chaînée .....	2
Par exemple :.....	2
Avantages des listes chaînées : .....	2
Désavantages des listes chaînées : .....	2
Comment fonctionne une liste chaînée ?.....	2
Un schéma pour mieux comprendre .....	2
Utilité des listes chaînées ? .....	2
Explication de l'application .....	3
Structures des listes utilisées .....	3
List_films .....	3
List_clients.....	3
Description des fonctions utilisées.....	3
Description des données : .....	6
AllandCambier01.dat :.....	6
AllandCambier02.dat.....	6
AllandCambier03.dat.....	6
Mode opératoire .....	6
Démarrage du programme .....	6
Menu client .....	7
Acheter une place pour film/avant-première/art et essai.....	7
Afficher les salles de cinéma .....	8
Menu administrateur .....	8

## Explication de la notion de liste chaînée

Contrairement à un vecteur qui est prévu pour contenir une quantité fixe de données, définie lors de sa déclaration, tout en occupant un espace mémoire fixe, une liste chaînée peut contenir une quantité non définie de données et adapte l'espace mémoire qu'elle occupe selon la quantité de données qu'elle contient.

Par exemple :

- une liste chaînée composée de 2 éléments occupera l'espace mémoire de 3 éléments (3ième éléments = NULL dans une liste unidirectionnelle)

- un vecteur de taille 24 contenant 2 éléments occupera l'espace mémoire de 24 éléments.

Avantages des listes chaînées :

- Allocation dynamique de la mémoire.

- Peut contenir un nombre illimité d'éléments.

Désavantages des listes chaînées :

- Lourd en syntaxe.

- notion pas évidente à prendre en main quand nouvelle.

Comment fonctionne une liste chaînée ?

Une liste chaînée fonctionne avec une structure dans laquelle se trouve un pointeur « next », du propre type de la structure.

Lorsqu'on déclare une liste, on l'initialise en lui donnant la valeur NULL, et chaque fois qu'on rajoute un élément à la liste, on alloue de la mémoire avec les fonctions malloc() et sizeof(StructDeLaListe) il ne faut pas oublier de préciser liste->next = NULL, le pointeur next pointer vers l'élément suivant .

Un schéma pour mieux comprendre

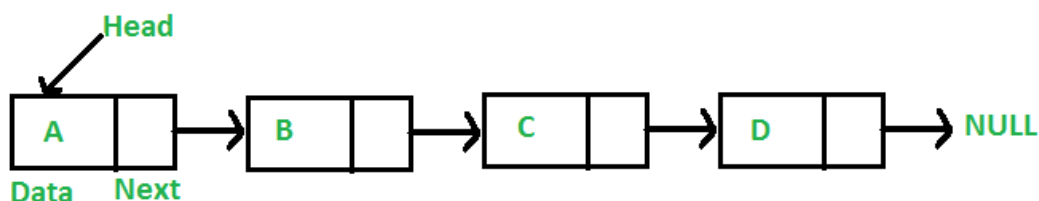


Figure 1 : Schéma d'une liste chaînée unidirectionnelle

Utilité des listes chaînées ?

Les listes chaînées sont utiles pour contenir une quantité de données non définie, et incertaine, susceptibles d'être ajoutées ou supprimées. Elles servent aussi à optimiser l'utilisation de mémoire vive d'un programme.

## Explication de l'application

Cette application concerne la gestion d'un cinéma :

L'utilisateur du programme pourra agir en tant que client ou administrateur du cinéma.

En tant que client : il pourra voir les films/avant-premières/art et essais programmés, le nombre de places restantes, le prix d'une place, la durée, si c'est en 3D ou pas, et à la suite de ça l'utilisateur aura le choix d'acheter une ou des place(s) pour un film.

Le client pourra aussi consulter les salles de cinéma, ainsi que leur capacité maximale.

L'administrateur pourra gérer les salles, les films, et les clients (il pourra en supprimer).

## Structures des listes utilisées

### List\_films

```
typedef struct List_films
{
    char titre[MAX_STRING_LENGTH];
    int type, duree, is3d, placesDispo, numSalle;
    float prix;
    struct List_films *next;
} Film, ListeFilms;
```

### List\_clients

```
typedef struct List_clients
{
    char nom[MAX_STRING_LENGTH], prenom[MAX_STRING_LENGTH];
    int nbPlaces, numFilm;
    struct List_clients *next;
} Client, ListeClients;
```

## Description des fonctions utilisées

```
Film *createFilm(char *titre, int is3d, int duree, int type, int places, float prix);
```

Crée un objet de type Film.

```
Client *createClient(char *nom, char *prenom, int nbPlaces, int numFilm);
```

Crée un objet de type Client.

```
ListeFilms *emptyFilmsList(void);
```

Renvoie une liste de typer ListeFilms vide.

```
int isEmptyFilmList(ListeFilms *L);
```

Détermine si une ListeFilms est vide.

```
int isEmptyClientsList(ListeClients *L);
```

Détermine si une ListeClients est vide.

```
long lenFilmsList(ListeFilms *L);
```

Retourne la taille d'une ListeFilms.

```
void filmsListPlaceInit(ListeFilms *L, int nbPl, int choix);
```

Calcul le nombre de places restantes pour les films, selon les clients déjà encodés.

```
int lenClientsList(ListeClients *L);
```

Retourne la taille d'une ListeClients.

```
Film *addFilmAt(Film *L, char *titre, int is3d, int duree, int type,  
int places, float prix, int pos);
```

Permet d'ajouter un film à une position précise d'une ListeFilms.

```
Client *addClientAt(Client *L, char *nom, char *prenom, int nbPlace,  
int numFilm, int pos);
```

Permet d'ajouter un client à une position précise d'une ListeClients.

```
void printFilmsListe(ListeFilms *L, int type);
```

Imprime la liste des films diffusés dans la console selon le type (1=films, 2=avant-premières, 3=arts et essais).

```
void replaceChar(char *str, char old, char new);
```

Remplace un caractère par un nouveau dans une chaîne de caractères.

```
void decalChar(char *str);
```

Décale la chaîne de caractère, si les premiers caractères sont des espaces.

```
void introduction(Salle *s, Client *cl, ListeFilms *f, FILE *clts);
```

Introduit le menu utilisateur des clients.

```
void tailleDesSalles(Salle *s);
```

Menu qui permet à l'utilisateur d'afficher les salles.

```
void ventePlace(Film *L, Client *cl, FILE *f);
```

Menu qui permet la vente de place(s) à l'utilisateur.

```
void acheterPlace(Film *L, Client *cl, int nbPl, int choixFilm, char  
*nom, char *prenom);
```

Fonction qui effectue l'achat de place(s).

```
float calculerPrix(int duree, int is3d);
```

Fonction qui calcule le prix d'une place pour un film.

```
void introAdmin(ListeClients *cl, ListeFilms *f, Salle *s);
```

Menu d'introduction pour les administrateurs.

```
void printClients(ListeClients *L);
```

Affiche la liste des clients dans la console.

```
ListeClients *freeClientsList(ListeClients *L);
```

Vide complètement une liste de clients.

```
float getPrixFilmAt(ListeFilms *L, int pos);
```

Retourne le prix d'un film situé dans une position précise d'une ListeFilms.

```
Client *freeClientAt(ListeClients *L, int pos);
```

Supprime un client à une position précise d'une ListeClients.

## Description des données :

### AllandCambier01.dat :

Contient les données liées aux films

Lecture : "%1d%1d%3d%2d%41s",&typeFilm,&is3d,&dureeFilm,&numsalle,&titreFilm

### AllandCambier02.dat

Contient les données relatives aux salles

Lecture : "%2d%3d",&s[i].num, &s[i].placesDispo

### AllandCambier03.dat

Contient les données des clients

Lecture : "%s %s %1d %2d", &prenom, &nom, &nbPlaces, &choixFilm

## Mode opératoire

### Démarrage du programme

```
*****
***** Bienvenue au Cinema *****
*****
1) Acceder en tant que client
2) Acceder en tant qu'administrateur
3) Quitter le programme

Votre choix ? (1-3)
```

- Afin d'accéder au menu des clients, il faut taper 1 puis « enter » dans la console.
- Afin d'accéder au menu des administrateurs, il faut taper 2 puis « enter » dans la console.
- Afin de quitter le programme, il faut taper 3 puis « enter » dans la console.

## Menu client

```
*****
***** Menu client *****
*****

Que voulez vous faire?
1) Voir les films diffuses en ce moment
2) Voir les avants-premieres qui auront lieus prochainement
3) Voir les evenements qui auront lieus prochainement
4) voir les salles disponibles
5) Retour

entrez le numero de votre choix :
```

-Afin de voir les films diffusés en ce moment, il faut taper 1 puis « enter » dans la console.

-Afin de voir les avant-premières diffusées en ce moment, il faut taper 2 puis « enter » dans la console.

-Afin de voir les arts et essais diffusés en ce moment, il faut taper 3 puis « enter » dans la console.

-Afin de voir les salles de cinéma disponibles, il faut taper 4 puis « enter » dans la console.

## Acheter une place pour film/avant-première/art et essai

```
Souhaitez vous acheter une/des place(s) pour un film? (Y/N)
y
Quel film voulez-vous regarder ? (entrez le numero du film)
29
Combien de places souhaitez-vous? (max 5 places)
1
confirmez-vous l'achat de 1 place(s) pour 15.90 euros? (Y/N)
y
Votre nom :
Cambier
Votre prenom :
Jean-Sebastien
```

Après avoir afficher les films, le programme vous propose d'acquérir des places de cinéma

-si vous en voulez, mettez « Y » dans la console, sinon mettez « n ».

-Entrer le numéro du film que vous voulez voir (voir liste des films proposés).

-entrer le nombre de place que vous désirez.

-confirmer l'achat en entrant « y », puis en entrant votre nom et prénom.



### Afficher les salles de cinéma

```
1) Afficher toutes les salles
2) Afficher les salles des petite capacite (130 places)
3) Afficher les salles Moyennes (200 places)
4) Afficher les grandes salles (350 places)
5) Quitter
```

-Tapez 1 pour toutes les salles.

-Tapez 2 pour les petites salles.

-Tapez 3 pour les salles moyennes.

-Tapez 4 pour les grandes salles.

-Tapez 5 pour retourner au menu client.

### Menu administrateur

```
*****
*****Menu administrateur*****
*****
1) Gerer les films
2) Gerer les clients
3) Gerer les salles
4) Retour

Votre choix? (1-4)
```

-Tapez 1 pour gérer les films.

-Tapez 2 pour gérer les clients.

-Tapez 3 pour gérer les salles.

-Tapez 4 pour retourner au menu principal.