





# Système de réservation de billets de cinéma

Sous l'encadrement :

Professeur.AL AMRANI Yassine



# NOTRE PLAN



Introduction



Méthodologie



Présentation générale



Conception et fonctionnalités



Contexte de l'étude



Conclusion



Problématique et objectif



# INTRODUCTION



# INTRODUCTION

Bienvenue à cette présentation de notre projet de programme de réservation de billets de cinéma en langage C. Le cinéma est une expérience captivante qui nous transporte dans des mondes imaginaires. Notre programme vise à simplifier la réservation de billets pour que vous puissiez profiter pleinement de cette expérience.

Plongeons ensemble dans les détails de ce projet passionnant où la programmation rencontre la magie du cinéma.



# Présentation générale



# Présentation générale

Notre projet vise à développer un programme informatisé permettant aux utilisateurs de réserver des billets pour divers films et séances dans les cinémas partenaires. Ce programme adopte une architecture modulaire et utilise des fichiers pour stocker les informations sur les films, les horaires des séances et les réservations des utilisateurs.





*A Night To Remember*

# Contexte de l'étude

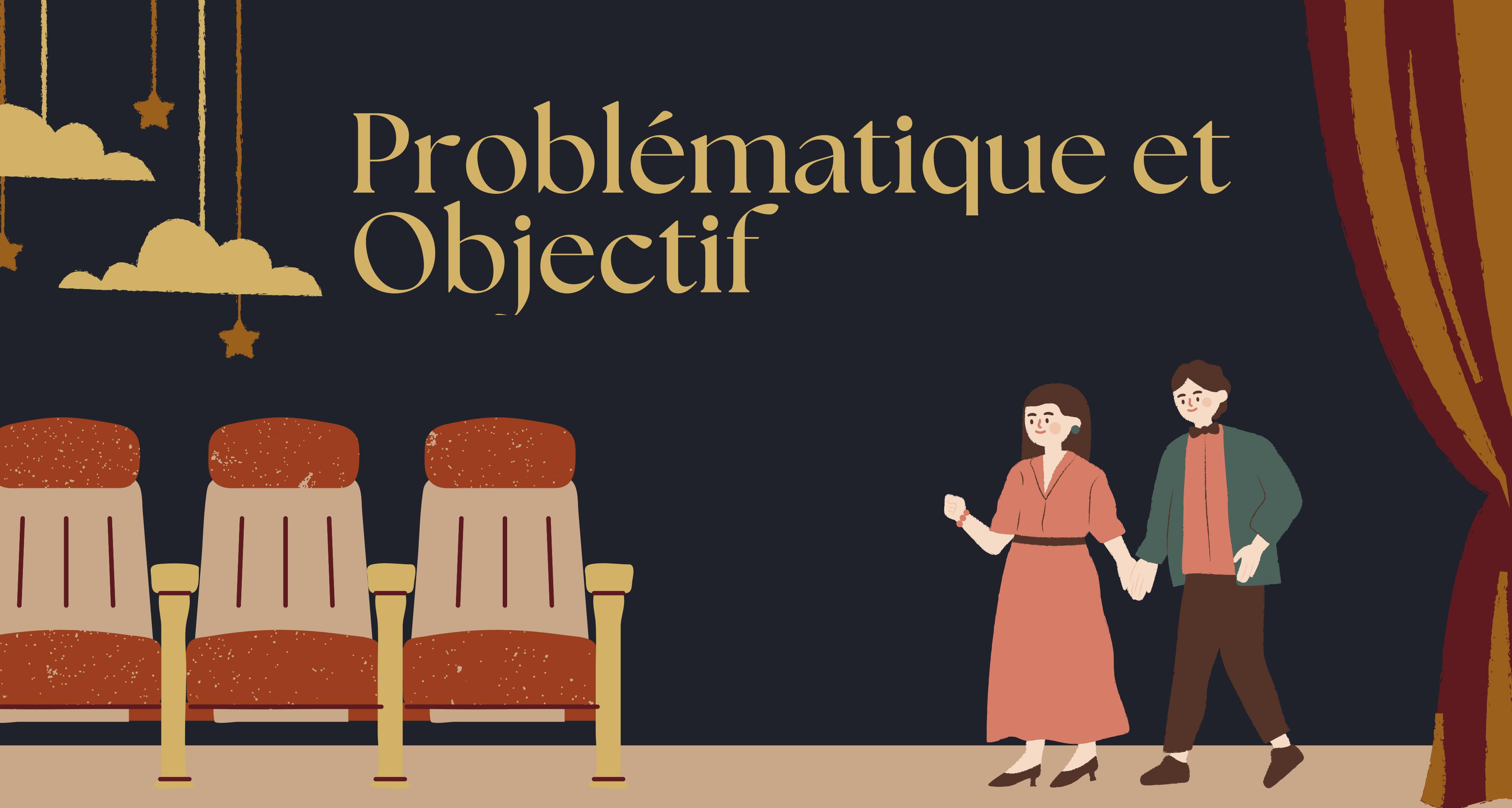


# Contexte de l'étude

A decorative graphic on the left side of the slide features a white movie ticket stub at the top. Below it is a dark blue, textured vertical band with white speckles, resembling a film strip or a movie ticket. The bottom part of the graphic consists of several stylized, overlapping curved shapes in white, gold, and dark blue, creating a sense of motion and depth.

Notre projet s'inscrit dans un marché cinématographique dynamique où les cinémas aspirent à moderniser leurs systèmes de réservation pour répondre aux attentes croissantes des clients. Ces attentes portent notamment sur la simplicité, la sécurité et l'ergonomie des interfaces. Ainsi, le développement d'un programme efficace de réservation de billets de cinéma devient essentiel pour offrir une expérience améliorée aux utilisateurs.

# Problématique et Objectif



# Problématique

- Obtention difficile des billets dans un court laps de temps et complexité du choix parmi les diverses séances peuvent dissuader les utilisateurs de réserver à l'avance.
- Gestion manuelle des réservations sujette à des erreurs et inefficace pour maintenir la mise à jour des séances et des disponibilités.
- Tarification uniforme des billets sans considération des tarifs différents pour les enfants et les adultes, rendant les billets inaccessibles à certains groupes de personnes.



# Objectif

- Développer un système de réservation de billets de cinéma en ligne simplifiant la recherche, la sélection et la réservation, offrant une expérience plus pratique.
- Mettre en place un système de gestion des séances en temps réel, permettant aux utilisateurs de vérifier facilement les horaires et la disponibilité.
- Intégrer une tarification dynamique tenant compte des différents tarifs pour garantir l'accessibilité des billets à un large public.

# Méthodologie



## Analyse des besoins #1

Cette phase consiste à identifier et à comprendre les besoins des utilisateurs ainsi que les exigences fonctionnelles et techniques du programme.

## Développement #3

Une fois la conception approuvée, le développement du programme démarre, en suivant les bonnes pratiques de programmation et en utilisant les technologies appropriées

## Conception #2

Cette étape implique la création de la conception globale du programme, y compris l'architecture système, les fonctionnalités principales.

## Tests et validation #4

Le programme est soumis à des tests approfondis pour vérifier son bon fonctionnement, sa sécurité et sa convivialité. Les retours des utilisateurs peuvent également être pris en compte à cette étape.

# Conception et fonctionnalités

## Conception

- Architecture Modulaire
- Encapsulation des Données
- Clarté et Cohérence

Partie 1

## fonctionnalités

- Complétude des Fonctionnalités
- Performance et Efficacité
- Facilité d'Utilisation et Convivialité

Partie 2

# Conception

## Architecture modulaire

Le code est conçu de manière modulaire, avec des modules distincts pour la définition des structures, la déclaration des fonctions et leur implémentation. Chaque module a un objectif clair et spécifique, ce qui facilite la gestion du code et améliore sa compréhension. Cette architecture modulaire favorise la réutilisabilité du code, permettant l'ajout de nouvelles fonctionnalités ou la modification de parties existantes sans perturber le système dans son ensemble.



## Encapsulation des Données

Les données sont encapsulées dans des structures appropriées pour les films, les séances et les clients, offrant une organisation claire et une lisibilité accrue du code. Cette encapsulation regroupe les données pertinentes ensemble, renforçant ainsi la sécurité et réduisant les risques de corruption des données.

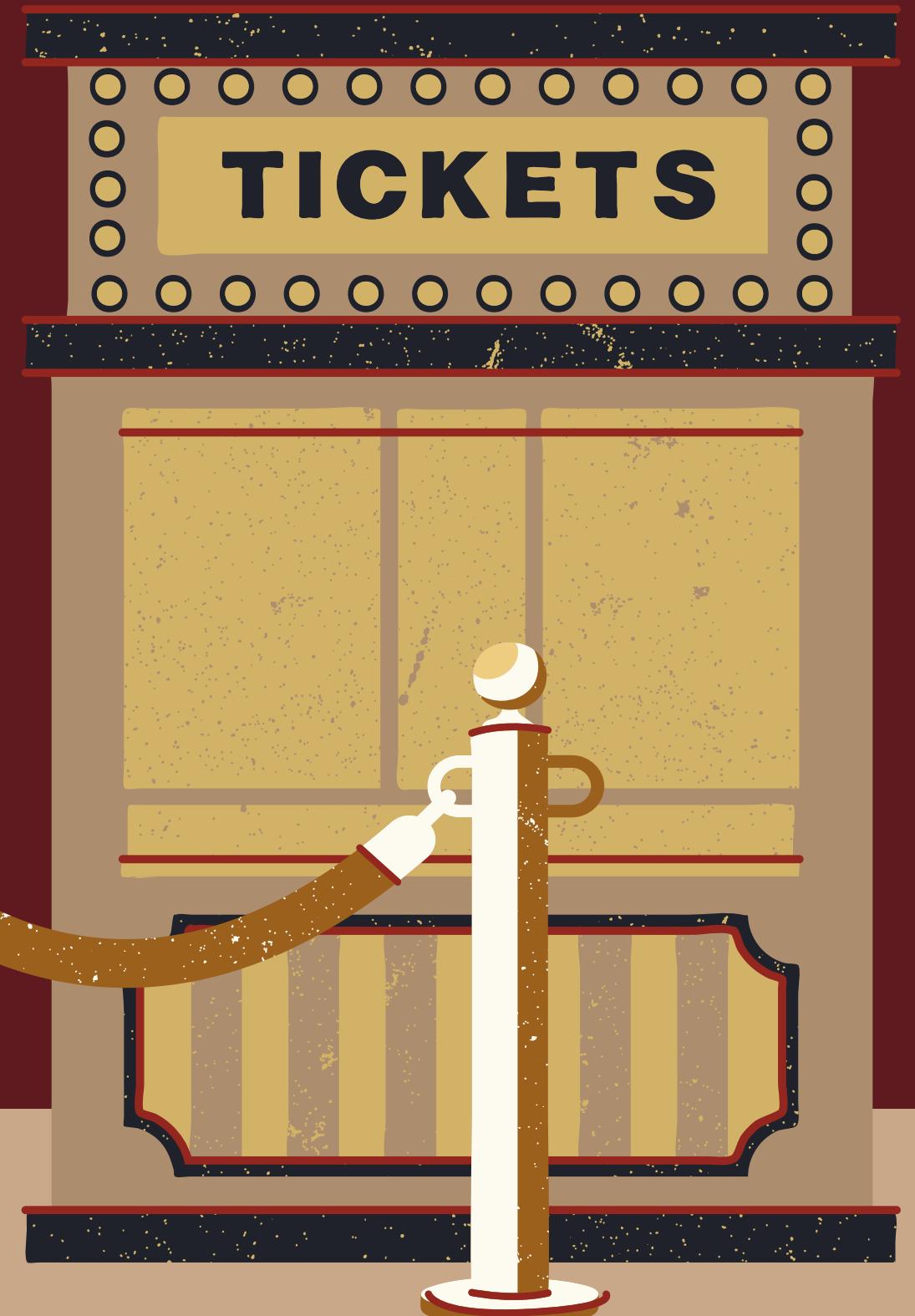
# Conception

## Clarté et Cohérence

Le code est structuré de manière claire et uniforme, respectant des conventions cohérentes. Cette cohérence garantit une expérience utilisateur homogène et prévisible, renforçant ainsi la convivialité et la satisfaction de l'utilisateur.



# Fonctionnalités



# Définition des Structures

Le programme utilise trois structures clés: **film**, **Seance** et **client**, pour simplifier la gestion des données et stocker de manière organisée les détails. Ces structures permettent également de fournir un service personnalisé aux clients en suivant leurs réservations et en conservant leurs informations.

## Implémentation des Fonctions :

Chaque fonction, implémentée est associée aux options des menus pour une utilisation

**Ajouter\_film()** , **Afficher\_films()**,  
**trouver\_film()**, **trouver\_seance()**,  
**reserver\_billet()** ,**modifier\_reservation()**,  
**liste\_des\_reservation()** ...

# Définition des Fonctions

Les fonctions dans notre programme jouent un rôle crucial en définissant les fonctionnalités essentielles. La fonction **menu()** gère l'interface utilisateur en proposant des choix entre le menu Employé et le menu Client. La fonction **menu\_employe()** permet aux employés de gérer les séances et les films. Quant à la fonction **menu\_client()**, elle facilite la réservation de billets et la consultation des films .

## Utilisation de fichiers :

Les détails des films et des séances sont conservés dans les fichiers texte (**data.txt** et **Seances.txt**), tandis que les réservations sont enregistrées dans **oldTransection.txt**. Les modifications temporaires sont stockées dans **tempTransection.txt** avant d'être intégrées définitivement.

# Fonctionnalités

## main.c

```
1 #include "Structures.h"
2 #include "Fonctions.h"
3 #include "../conio2.h"
4
5 #define TEXT_COLOR BLUE
6 #define BACKGROUND_COLOR WHITE
7 #define MENU_BORDER_COLOR RED
8
9 int main()
10 {
11     textcolor(TEXT_COLOR);
12     textbackground(BACKGROUND_COLOR);
13     menu();
14 }
```

## Explications

### Main.c

Ce code contient la fonction **main()**, qui est le point d'entrée de notre programme.

**Structures.h**: inclut le fichier d'en-tête contenant les définitions de structures.

**Fonctions.h**: inclut le fichier d'en-tête contenant les prototypes de fonctions

**Conio2h**: Permet l'utilisation de fonctions windows pour la gestion de la console.

### Fonction main()

Initialise les variables nécessaires à la gestion des attributs de la console

Affiche le menu principal en appelant la fonction **menu()**  
Restaure les attributs de la console après l'exécution du programme

## Structures

```
1 #ifndef STRUCTURES_H
2 #define STRUCTURES_H
3
4 typedef struct
5 {
6     char code[20];
7     char nom[20];
8     char genre[20];
9     int prix;
10 }film;
11
12 typedef struct {
13     char code_film[20];
14     int Num;
15     char horaire[20];
16     int salle;
17     int nombre_places_disponibles;
18 } Seance;
19
20 typedef struct{
21     int Numero_ticket;
22     char nom_film[20];
23     char nom[20];
24     int age;
25     int telephone;
26     int Num;
27     int montant;
28 }client;
29
30#endif
```

## Explications

### Structures:

Ce fichier contient les définitions des structures de données utilisées dans le projet :

#### Structure film :

Stocke les informations sur un film, telles que le code, le nom, le genre et le prix.

#### Structure séance

Stocke les informations sur une séance, telles que le code du film, le numéro de séance, l'horaire, la salle et le nombre de places disponibles.

# Fonctionnalités

## fonctions.h

```
1 #ifndef FONCTIONS_H
2 #define FONCTIONS_H
3
4 void menu();
5 void menu_employe();
6 void menu_client();
7 void afficher_genres();
8 void Ajouter_film();
9 void Ajouter_Seance();
10 void Afficher_films();
11 void Afficher_seance();
12 void trouver_film();
13 void trouver_seance(char code_film[20]);
14 void reserver_billet();
15 void liste_des_reservations();
16 void modifier_reservation();
17 void attendre_entree();
18
19 #endif
20
```

## Explications

Le fichier fonctions.h contient les déclarations de toutes les fonctions utilisées dans le programme. Ces déclarations permettent aux autres parties du code d'utiliser ces fonctions sans avoir besoin de connaître leur implémentation détaillée. Le fichier inclut également des protections contre les inclusions multiples pour éviter les erreurs de duplication, ainsi que des commentaires pour expliquer brièvement le but de chaque fonction.

# Fonctionnalités

void menu()

```
1 #include "Structures.h"
2 #include "Fonctions.h"
3
4 void menu() {
5     int choix_principal;
6     do {
7         system("cls");
8
9         printf("\n=====");
10        printf("\n");
11        printf("\t Reservation de billets de cinéma ");
12        printf("\n");
13        printf("\n=====");
14
15        printf("\nAppuyez sur 1 pour Menu Employé");
16        printf("\nAppuyez sur 2 pour Menu Client");
17        printf("\nAppuyez sur 0 pour Quitter ");
18        scanf("%d", &choix_principal);
19
20        switch (choix_principal) {
21            case 1:
22                menu_employe();
23                break;
24            case 2:
25                menu_client();
26                break;
27            case 0:
28                exit(0);
29                break;
30            default:
31                printf("Choix incorrect !");
32                break;
33        }
34    } while (choix_principal != 0);
35 }
```

Explications

Fonction menu() :  
Affiche un menu principal avec des options pour l'employé et le client.  
Permet à l'utilisateur de choisir une option et appelle la fonction correspondante

# Fonctionnalités

## void menu\_employe()

```
37 void menu_employe()
38 {
39     int choix_employe;
40
41     do {
42         printf("\n\nMenu Employe\n");
43         printf("Appuyez sur 1 pour Ajouter un film\n");
44         printf("Appuyez sur 2 pour Ajouter une seance\n");
45         printf("Appuyez sur 3 pour Afficher la liste des films\n");
46         printf("Appuyez sur 4 pour Rechercher un film\n");
47         printf("Appuyez sur 5 pour Afficher les seance disponibles\n");
48         printf("Appuyez sur 6 pour Afficher les anciens reservations\n");
49         printf("Appuyez sur 0 pour Retourner au menu principal\n");
50
51         printf("Entrez votre choix : ");
52         scanf("%d", &choix_employe);
53
54         switch (choix_employe)
55     {
56             case 1:
57                 Ajouter_film();
58                 attendre_entree();
59                 system("cls");
60                 break;
61             case 2:
62                 Ajouter_Seance();
63                 attendre_entree();
64                 system("cls");
65                 break;
66             case 3:
67                 Afficher_films();
68                 attendre_entree();
69                 system("cls");
70                 break;
71             case 4:
72                 trouver_film();
73                 attendre_entree();
74
75                 attendre_entree();
76                 system("cls");
77                 break;
78             case 5:
79                 Afficher_seance();
80                 attendre_entree();
81                 system("cls");
82                 break;
83             case 6:
84                 liste_des_reservations();
85                 attendre_entree();
86                 system("cls");
87                 break;
88             case 0:
89                 system("cls");
90                 menu();
91                 break;
92             default:
93                 printf("Choix incorrect !");
94                 break;
95         }
96     } while (choix_employe != 0);
```

## Explications

### Fonction menu():

La fonction menu-employe() affiche un menu pour les employés du système de réservation de billets de cinéma. L'employé peut choisir parmi plusieurs options telles que l'ajout de film, la visualisation des films disponibles, la recherche d'un film spécifique, l'affichage des anciennes réservations, l'ajout d'une séance et l'affichage des séances disponibles. La fonction utilise une boucle do-while pour permettre à l'employé de sélectionner plusieurs options jusqu'à ce qu'il choisisse de retourner au menu principal en sélectionnant l'option 0.

menu principal avec des options pour l'employé et le client.  
Permet à l'utilisateur de choisir une option et appelle la fonction correspondante

# Fonctionnalités

## void menu\_client()

```
155  void menu_client() {
156      int choix_client;
157      do {
158          afficher_menu_client();
159          scanf("%d", &choix_client);
160
161          switch (choix_client) {
162              case 1:
163                  reserver_billet();
164                  attendre_entree();
165                  break;
166              case 2:
167                  Afficher_films();
168                  attendre_entree();
169                  break;
170              case 3:
171                  trouver_film();
172                  attendre_entree();
173                  break;
174              case 4:
175                  modifier_reservation();
176                  attendre_entree();
177                  break;
178              case 0:
179                  menu();
180                  break;
181              default:
182                  printf("Choix incorrect !\n");
183                  attendre_entree();
184                  break;
185          }
186      } while (choix_client != 0);
187 }
```

## Explications

La fonction menu-client() affiche un menu pour les clients, leur permettant de choisir parmi plusieurs options telles que la réservation de billets, la visualisation des films disponibles, la recherche d'un film spécifique et la modification d'une réservation existante.

Elle utilise une boucle do-while pour permettre à l'utilisateur de sélectionner plusieurs options jusqu'à ce qu'il choisisse de revenir au menu principal.

# Fonctionnalités

## void Ajouter\_film()

```
139
140 void Ajouter_film() {
141     FILE *fp;
142     film f;
143     int choix_genre;
144
145     printf("Entrez le code du film: ");
146     scanf("%s", f.code);
147     printf("Entrez le nom : ");
148     scanf("%s", f.nom);
149     printf("Choisissez le genre du film:");
150     afficher_genres();
151     printf("Entrez le numero correspondant : ");
152     scanf("%d", &choix_genre);
153
154     switch(choix_genre) {
155         case 1:
156             strcpy(f.genre, "Aventure");
157             break;
158         case 2:
159             strcpy(f.genre, "Action");
160             break;
161         case 3:
162             strcpy(f.genre, "Comedie");
163             break;
164         case 4:
165             strcpy(f.genre, "Drame");
166             break;
167         case 5:
168             strcpy(f.genre, "Romance");
169             break;
170         case 6:
171             strcpy(f.genre, "Science-Fiction");
172             break;
173         case 7:
174             strcpy(f.genre, "Documentaire");
175             break;
176         case 8:
177             strcpy(f.genre, "Mystere");
178             break;
179         case 9:
180             strcpy(f.genre, "Autre");
181             break;
182         default:
183             printf("Genre non valide !");
184             return;
185     }
```

```
186
187     printf("Entrez le prix du billet: ");
188     scanf("%d", &f.prix);
189
190     fp = fopen("data.txt", "a+");
191     if (fp == NULL) {
192         printf("Fichier non trouve");
193     } else {
194         fprintf(fp, "%s %s %s %d\n", f.code, f.nom, f.genre, f.prix);
195         printf("Enregistrement insere avec succes");
196     }
197     printf("\n");
198     fclose(fp);
199 }
```

## Explications

La fonction Ajouter-film() permet à l'utilisateur d'ajouter un nouveau film dans le système de réservation de billets de cinéma. L'utilisateur saisit le code, le nom, le genre et le prix du billet pour le nouveau film. Le genre est choisi en sélectionnant un numéro correspondant à partir d'une liste affichée à l'écran. Les informations du film sont ensuite enregistrées dans un fichier texte nommé "data.txt". Un message de confirmation est affiché si l'opération réussit.

# Fonctionnalités

## void Ajouter\_Seance()

```
268 void Ajouter_Seance() {
269     FILE *fp;
270     Seance s;
271
272     printf("Entrez le code du film pour cette seance: ");
273     scanf("%s", s.code_film);
274     printf("Entrez le numero de seance : ");
275     scanf("%d", &s.Num);
276     printf("Choisissez l'horaire de cette seance (jour heure) : ");
277     scanf("%s", s.horaire);
278     printf("Entrez le numero de salle : ");
279     scanf("%d", &s.salle);
280     printf("Entrez le nombre de places disponibles : ");
281     scanf("%d", &s.nombre_places_disponibles);
282
283     fp = fopen("Seances.txt", "a+");
284     if (fp == NULL) {
285         printf("Fichier non trouve");
286         return;
287     } else {
288         fprintf(fp, "%s %d %s %d %d\n", s.code_film, s.Num, s.horaire, s.salle, s.nombre_places_disponibles);
289         printf("Enregistrement insere avec succes");
290     }
291     printf("\n");
292     fclose(fp);
293 }
```

## Explications

La fonction Ajouter-Seance() permet d'ajouter une nouvelle séance pour un film dans le système de réservation de billets de cinéma. L'utilisateur saisit les détails de la séance, tels que le code du film, le numéro de séance, l'horaire, le numéro de salle et le nombre de places disponibles. Ces informations sont ensuite enregistrées dans un fichier texte nommé "Seances.txt". Un message de confirmation est affiché si l'opération réussit.

# Fonctionnalités

## Ajouter-film() & afficher-genres()

```
139
140 void Ajouter_film() {
141     FILE *fp;
142     film f;
143     int choix_genre;
144
145     printf("Entrez le code du film: ");
146     scanf("%s", f.code);
147     printf("Entrez le nom : ");
148     scanf("%s", f.nom);
149     printf("Choisissez le genre du film:");
150     afficher_genres();
151     printf("Entrez le numero correspondant : ");
152     scanf("%d", &choix_genre);
153
154     switch(choix_genre) {
155         case 1:
156             strcpy(f.genre, "Aventure");
157             break;
158         case 2:
159             strcpy(f.genre, "Action");
160             break;
161         case 3:
162             strcpy(f.genre, "Comedie");
163             break;
164         case 4:
165             strcpy(f.genre, "Drame");
166             break;
167         case 5:
168             strcpy(f.genre, "Romance");
169             break;
170         case 6:
171             strcpy(f.genre, "Science-Fiction");
172             break;
173         case 7:
174             strcpy(f.genre, "Documentaire");
175             break;
176         case 8:
177             strcpy(f.genre, "Mystere");
178             break;
179         case 9:
180             strcpy(f.genre, "Autre");
181             break;
182         default:
183             printf("Genre non valide !");
184             return;
185     }
```

```
186     printf("Entrez le prix du billet: ");
187     scanf("%d", &f.prix);
188
189     fp = fopen("data.txt", "a+");
190     if (fp == NULL) {
191         printf("Fichier non trouve");
192     } else {
193         fprintf(fp, "%s %s %s %d\n", f.code, f.nom, f.genre, f.prix);
194         printf("Enregistrement insere avec succes");
195     }
196     printf("\n");
197     fclose(fp);
198 }
199
200 void afficher_genres() {
201     printf("\nListe des genres de films :\n");
202     printf("1. Aventure\n");
203     printf("2. Action\n");
204     printf("3. Comedie\n");
205     printf("4. Drame\n");
206     printf("5. Romance\n");
207     printf("6. Science-Fiction\n");
208     printf("7. Documentaire\n");
209     printf("8. Mystere\n");
210     printf("9. Horreur\n");
211     printf("10. Autre\n");
212 }
```

## Explications

La fonction Ajouter-film() permet à l'utilisateur d'ajouter un nouveau film dans le système de réservation de billets de cinéma. L'utilisateur saisit le code, le nom, le genre et le prix du billet pour le nouveau film.

Le genre est choisi en sélectionnant un numéro correspondant à partir d'une liste affichée à l'écran grâce au fonction afficher-genres.

Cette fonction est utilisée pour présenter à l'utilisateur les options disponibles lorsqu'il doit choisir un genre de film.

Les informations du film sont ensuite enregistrées dans un fichier texte nommé "data.txt". Un message de confirmation est affiché si l'opération réussit.

# Fonctionnalités

## void Afficher\_film() & void afficher-seance()

```
295 void Afficher_films() {
296     film f;
297     FILE *fp;
298     fp = fopen("data.txt", "r");
299     if (fp == NULL) {
300         printf("Fichier introuvable !");
301         exit(1);
302     } else {
303         printf("\nListe de tous les films :\n");
304         while (fscanf(fp, "%s %s %s %d", f.code, f.nom, f.genre, &f.prix) != EOF) {
305             printf("\n\tCode : %s", f.code);
306             printf("\n\tNom du film : %s", f.nom);
307             printf("\n\tGenre : %s", f.genre);
308             printf("\n\tPrix du billet : %d$\n", f.prix);
309         }
310     }
311     fclose(fp);
312 }
313 void Afficher_seance() {
314     Seance S;
315     FILE *fp;
316     fp = fopen("Seances.txt", "r");
317     if (fp == NULL) {
318         printf("Fichier introuvable !");
319         exit(1);
320     } else {
321         printf("\nListe de tous les seances :\n");
322         while (fscanf(fp, "%s %d %s %d %d", S.code_film, &S.Num, S.horaire, &S.salle, &S.nombre_places_disponibles) != EOF) {
323             printf("\n\tCode de film : %s", S.code_film);
324             printf("\n\tNumero de seance : %d", S.Num);
325             printf("\n\tL'heure : %s", S.horaire);
326             printf("\n\tLa salle : %d", S.salle);
327             printf("\n\tLa nombre des places disponibles : %d$\n", S.nombre_places_disponibles);
328         }
329     }
330     fclose(fp);
331 }
```

## Explications

Les fonctions `Afficher-films()` et `Afficher-seance()` affichent respectivement la liste de tous les films et de toutes les séances disponibles. Elles ouvrent les fichiers correspondants en mode lecture, parcouruent les données ligne par ligne, et affichent les détails de chaque film ou séance. Si un fichier n'est pas trouvé, un message d'erreur est affiché et le programme se termine. Enfin, elles ferment les fichiers après avoir terminé leur traitement.

# Fonctionnalités

## Trouver\_film() & trouver\_seance()

```
269 void trouver_film()
270 {
271     film f;
272     FILE *fp;
273     char code_film[20];
274     char ch;
275     printf("Entrez le code du film : ");
276     scanf("%s", code_film);
277
278     fp = fopen("data.txt", "r");
279     if (fp == NULL)
280     {
281         printf("Fichier introuvable !");
282         exit(1);
283     }
284     else
285     {
286         while (fscanf(fp, "%s %s %s %d", f.code, f.nom, f.genre, &f.prix) != EOF)
287         {
288             if (strcmp(f.code, code_film) == 0)
289             {
290                 printf("\n Enregistrement trouvé\n");
291                 printf("\n\t\tCode : %s", f.code);
292                 printf("\n\t\tNom du film : %s", f.nom);
293                 printf("\n\t\tGenre : %s", f.genre);
294                 printf("\n\t\tPrix du billet : %d", f.prix);
295                 break;
296             }
297         }
298     }
299     fclose(fp);
300 }
```

```
366 void trouver_seance(char code_film[20]) {
367     Seance s;
368     FILE *fp;
369
370     fp = fopen("Seances.txt", "r");
371     if (fp == NULL) {
372         printf("Fichier introuvable !");
373         exit(1);
374     }
375     else {
376         int found = 0;
377         while (fscanf(fp, "%s %d %s %d", s.code_film, &s.Num, s.horaire, &s.salle, &s.nombre_places_disponibles) != EOF) {
378             if (strcmp(s.code_film, code_film) == 0) {
379                 found = 1;
380                 printf("\nLes séances disponibles pour ce film:\n");
381                 printf("\n\tCode de film : %s", s.code_film);
382                 printf("\n\tNuméro de séance : %d", s.Num);
383                 printf("\n\tL'horaire : %s", s.horaire);
384                 printf("\n\tLa salle : %d", s.salle);
385                 printf("\n\tLa nombre des places disponibles : %d\n", s.nombre_places_disponibles);
386             }
387         }
388         if (!found) {
389             printf("Aucune séance correspondant au film sélectionné !\n");
390         }
391     }
392 }
393 }
```

## Explications

Trouver-film() ouvre le fichier des films, parcourt les enregistrements pour trouver celui correspondant au code donné, puis affiche ses détails s'il est trouvé.

Trouver-seance() ouvre le fichier des séances, parcourt les enregistrements pour trouver ceux correspondant au code donné, puis les affiche s'ils sont trouvés

# Fonctionnalités

## void reserver\_billet()

```
394 void reserver_billet()
395 {
396     film f;
397     Seance s;
398     Client c;
399     FILE *fp;
400     FILE *ufp;
401     FILE *sfp;
402     int numero_ticket;
403     char code_film[20];
404     char num_seance[10];
405     char ch;
406
407     fp = fopen("data.txt", "r");
408     if (fp == NULL)
409     {
410         printf("Fichier introuvable !");
411         exit(1);
412     }
413
414     printf("\n Pour Reserver un billet, Choisissez un film \n");
415     Afficher_films();
416     printf("\n Entrez le code du film que vous voulez : ");
417     scanf("%s", code_film);
418     printf("\n Voici les seances disponibles pour ce film: ");
419     trouver_seance(code_film);
420     printf("\n Entrez le numero de la seance que vous voulez : ");
421     scanf("%d", &c.Num);
422
423     sfp = fopen("Seances.txt", "r+");
424     if (sfp == NULL)
425     {
426         printf("Fichier introuvable !");
427         exit(1);
428     }
429     else
430     {
```

## Explications

- Pour réserver un billet, la fonction reserver-billet() :
  - Affiche les films disponibles et demande au client de choisir en entrant le code du film.
  - Affiche les séances disponibles pour ce film et demande au client de choisir une séance en entrant son numéro.
  - Vérifie la disponibilité de la séance choisie.
  - Vérifie le nombre des places restants dans la salle pendant cette seance

# Fonctionnalités

## void reserver\_billet()

```
431 int trouve = 0;
432 while (fscanf(sfptr, "%s %d %s %d %d", &S.code_film, &S.Num, &S.horaire, &S.salle, &S.nombre_places_disponibles) != EOF)
433 {
434     if (S.Num == C.Num && strcmp(S.code_film, code_film) == 0)
435     {
436         if (S.nombre_places_disponibles <= 0) {
437             printf("Désolé, il n'y a plus de places disponibles pour cette séance.\n");
438             return;
439         }
440         printf("Remplissez les informations pour réserver\n");
441         printf("Votre nom : ");
442         scanf("%s", C.nom);
443         printf("Votre âge : ");
444         scanf("%d", &C.age);
445         printf("Numéro de téléphone : ");
446         scanf("%d", &C.telephone);
447         S.nombre_places_disponibles--;
448         fseek(sfptr, -1 * sizeof(Seance), SEEK_CUR);
449         fprintf(sfptr, "%s %d %s %d %d\n", S.code_film, S.Num, S.horaire, S.salle, S.nombre_places_disponibles);
450         break;
451     }
452 }
453 }

454 while (fscanf(fp, "%s %s %s %d", f.code, f.nom, f.genre, &f.prix) != EOF)
455 {
456     if (strcmp(f.code, code_film) == 0)
457     {
458         break;
459     }
460 }
461 fclose(fp);
462 if (C.age < 15) {
463     printf("Réduction de 30% est appliquée !\n");
464     f.prix = f.prix - (30 * f.prix / 100);
465 }

466
467 C.montant = f.prix;
468 numero_ticket++;
469 C.Numero_ticket = numero_ticket;
470 strcpy(C.nom_film, f.nom);

471 printf("Votre réservation est enregistrée avec succès\n\n");
472 printf("\n*****");
473 printf("\n***** Ticket de réservation *****");
474 printf("\n Numéro de ticket : %d", C.Numero_ticket);
475 printf("\n Nom : %s", C.nom);
476 printf("\n Âge : %d", C.age);
477 printf("\n Numéro de téléphone : %d", C.telephone);
478 printf("\n Nom du film : %s", C.nom_film);
479 printf("\n Prix par billet : %d $\n", C.montant);
480 printf("\n*****");

481 ufp = fopen("oldTransaction.txt", "a");
482 if (ufp == NULL)
483 {
484     printf("Fichier non trouvé");
485 }
486 else
487 {
488     fprintf(ufp, "%d %s %d %d %s %d \n", C.Numero_ticket, C.nom, C.age, C.telephone, C.nom_film, C.montant);
489 }
490 printf("\n");
491 fclose(ufp);
492 fclose(sfptr);
```

## Explications

Récupère les détails du film choisi.

Demande au client de saisir ses informations personnelles.

Applique une réduction de 30% pour les enfants en utilisant l'âge saisi par l'utilisateur,

Affiche un ticket de réservation avec les détails de la réservation.

Enregistre les détails de la réservation dans un fichier pour archiver l'historique des transactions.

# Fonctionnalités

## void liste\_des\_reservations

```
434 void liste_des_reservations()
435 {
436     char ch;
437     FILE *fp;
438
439     system("cls");
440
441     fp = fopen("oldTransection.txt", "r");
442     if (fp == NULL)
443     {
444         printf("Il n y a aucune reservation !");
445         exit(1);
446     }
447     else
448     {
449         while ((ch = fgetc(fp)) != EOF)
450             printf("%c", ch);
451     }
452     fclose(fp);
453     attendre_entree();
454 }
```

### Explications

liste-des-reservations() : c'est une option pour l'employée pour Affiche la liste des réservations présentes dans un fichier, s'il y en a, en les lisant et les affichant à l'écran.

# Fonctionnalités

## Void modifier\_reservation

```
518 void modifier_reservation() {
519     FILE *ufp, *tfp;
520     Client C;
521     int numero_ticket;
522     int choix;
523
524     printf("Entrez le numero de ticket de la reservation a modifier : ");
525     scanf("%d", &numero_ticket);
526
527     ufp = fopen("oldTransection.txt", "r");
528     if (ufp == NULL) {
529         printf("Fichier introuvable !");
530         return;
531     }
532
533     tfp = fopen("tempTransection.txt", "w");
534     if (tfp == NULL) {
535         printf("Erreur lors de l'ouverture du fichier temporaire !");
536         fclose(ufp);
537         return;
538     }
539
540     while (fscanf(ufp, "%d %s %d %s %d \n", &C.Numero_ticket, C.nom, &C.age, &C.telephone, C.nom_film, &C.montant) != EOF){
541         if (numero_ticket == C.Numero_ticket) {
542             printf("\n***** Details de la reservation *****\n");
543             printf("\n Numero de ticket : %d", C.Numero_ticket);
544             printf("\n 1. Nom : %s", C.nom);
545             printf("\n 2. Age : %d", C.age);
546             printf("\n 3. Numero de telephone : %d", C.telephone);
547             printf("\n 4. Nom du film : %s", C.nom_film);
548             printf("\n 5. Prix par billet : %d $", C.montant);
549             printf("\n***** Modification de la reservation *****\n");
550             printf("Entrez le numero de l'information a modifier : ");
551             scanf("%d", &choix);
552
553             switch (choix) {
554                 case 1:
555                     printf("Entrez le nouveau nom : ");
556                     scanf("%s", C.nom);
557                     break;
558
559                 case 2:
560                     printf("Entrez le nouvel age : ");
561                     scanf("%d", &C.age);
562                     break;
563
564                 case 3:
565                     printf("Entrez le nouveau numero de telephone : ");
566                     scanf("%d", &C.telephone);
567                     break;
568
569                 case 4:
570                     printf("Entrez le nouveau nom du film : ");
571                     scanf("%s", C.nom_film);
572                     break;
573
574                 case 5:
575                     printf("Entrez le nouveau prix par billet : ");
576                     scanf("%d", &C.montant);
577                     break;
578
579                 default:
580                     printf("Choix incorrect !");
581                     break;
582             }
583
584             fprintf(tfp, "%d %s %d %s %d \n", C.Numero_ticket, C.nom, C.age, C.telephone, C.nom_film,C.montant);
585             printf("\nLa reservation a ete mise a jour avec succes !");
586         } else {
587             printf("\nAucune reservation trouee avec le numero de ticket fourni !");
588         }
589     }
590     fclose(ufp);
591     fclose(tfp);
592
593     remove("oldTransection.txt");
594     rename("tempTransection.txt", "oldTransection.txt");
```

## Explications

Modifier-reservation() : Permet de modifier une réservation en demandant à l'utilisateur de saisir le numéro de ticket de la réservation à modifier, et le numéro de informations à modifier par saisie d'un numéro à partir de la liste affiché puis en mettant à jour les détails de cette réservation dans le fichier de réservations. Si aucune réservation correspondante n'est trouvée, un message approprié est affiché

# CONCLUSION

En conclusion, notre projet de gestion de cinéma offre une expérience utilisateur fluide et efficace. Grâce à une conception modulaire et une encapsulation appropriée des données, les fonctionnalités telles que la réservation de billets, l'affichage des films et la modification des réservations sont facilement accessibles et conviviales. Avec une architecture claire et cohérente, notre système est à la fois robuste et évolutif, offrant ainsi une solution complète pour la gestion d'un cinéma.





MERCI POUR VOTRE  
ATTENTION

