

Projet Technique de Programmation : PolyMusic

I. Présentation :

Il s'agit de produire un programme (fichiers source et un exécutable testé et opérationnel), dont le sujet est présenté dans la suite du document. Le développement de ce programme se fera en langage C.

A. Objectif du projet

- Mettre en œuvre les notions du cours de C
- Aborder des notions d'architecture logicielle
- Développer un regard critique dans l'implémentation
- Organiser son implémentation et avoir une vision globale du travail à réaliser
- Définir des interfaces
- Intégrer une librairie externe
- Gestion de mémoire
- Gestion de fichier
- Mise en œuvre de listes chaînées

B. Organisation

Le projet est individuel.

Le projet est pensé pour être réalisable durant les TP du cours de « Technique de Programmation ».

C. Ce qu'il faut rendre

- Le programme source complet largement commenté
- Un exécutable testé et opérationnel
- Présentation oral du travail effectué de 10 minutes lors de la dernière séance :
 - Organisation du programme: découpage en fonctions, rôle de ces fonctions, explications du programme
 - Mode d'emploi du programme
 - Bilan qualitatif du travail, difficultés rencontrées, etc.

Le tout sera placé dans un fichier archive à m'envoyer par mail.

D. Evaluation du projet

Elle repose sur les éléments suivants:

- Le programme source:
 - Respect de l'énoncé
 - Qualité technique du programme: découpage en fonctions, instructions, algorithmes, efficacité, gestion des erreurs...
 - Présentation du programme: indentation, commentaires et nommage
- Présentation oral du travail :
 - Organisation du programme et son mode d'emploi
 - Démonstration du programme
 - Interrogation individuelle sur le travail réalisé
 - Bilan

II. Présentation du projet : PolyMusic

A. Description

Le programme PolyMusic doit permettre de gérer une bibliothèque de chansons au format mp3 et de pouvoir les lire.

Pour obtenir la liste des mp3 disponibles, PolyMusic accède à une base de données. Cette base de données est un fichier qui contient les informations et le chemin des mp3.

PolyMusic dispose de 2 modes :

- Un mode utilisateur qui permet de chercher et de lire des musiques
- Un mode administrateur qui permet de modifier la liste des mp3 (et donc le fichier base de données associée)

Pour lire des mp3, je propose d'utiliser la library open source FMOD.

Pour l'obtenir, il faut créer un compte sur le site de FMOD.

Pour faire plus simple, je la diffuserai en cours.

PolyMusic fonctionne en fenêtre de commande.

Toutefois, il est possible de l'implémenter en utilisant une interface graphique.

Dans ce cas, je propose d'utiliser la library open source SDL.

Voici un site qui propose des chansons à télécharger qui sont libres de droit :

<http://freemusicarchive.org/>

B. Gestion du fichier de base de données

Le fichier de base de données est constitué de la liste des mp3 disponibles.

Pour chaque mp3, on doit retrouver les informations suivantes :

- Un ID unique
- Nom de l'artiste
- Nom de la chanson
- Album
- Style
- Chemin absolu du fichier mp3 (ex : C:\MyMusic\Artiste_Chanson.mp3)

Je propose que le fichier soit du format suivant :

```
1 ID1>Artiste1→Chanson1→Album1→Style1→Chemin1CRLE
2 ID2>Artiste2→Chanson2→Album2→Style2→Chemin2CRLE
3 ID3>Artiste3→Chanson3→Album3→Style3→Chemin3CRLE
4 ID4>Artiste4→Chanson4→Album4→Style4→Chemin4CRLE
5 ID5>Artiste5→Chanson5→Album5→Style5→Chemin5CRLE
6 ID6>Artiste6→Chanson6→Album6→Style6→Chemin6CRLE
7 ID7>Artiste7→Chanson7→Album7→Style7→Chemin7CRLE
8 ID8>Artiste8→Chanson8→Album8→Style8→Chemin8CRLE
9 ID9>Artiste9→Chanson9→Album9→Style9→Chemin9CRLE
```

- Une chanson par ligne
- Une tabulation pour séparer les éléments.
- L'ID correspond au numéro de la ligne

Contrainte :

Le fichier doit être trié en fonction du nom de l'artiste par ordre alphabétique croissant.

Conseil : Utiliser strcmp de la library <string.h> pour trier

Attention à la casse (minuscule / majuscule) !

Rappel du tableau ASCII :

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32;	Space	64	40	100	##64;	@	96	60	140	##96;	`
1	1	001	SOH (start of heading)	33	21	041	##33;	!	65	41	101	##65;	A	97	61	141	##97;	a
2	2	002	STX (start of text)	34	22	042	##34;	"	66	42	102	##66;	B	98	62	142	##98;	b
3	3	003	ETX (end of text)	35	23	043	##35;	#	67	43	103	##67;	C	99	63	143	##99;	c
4	4	004	EOT (end of transmission)	36	24	044	##36;	\$	68	44	104	##68;	D	100	64	144	##100;	d
5	5	005	ENQ (enquiry)	37	25	045	##37;	%	69	45	105	##69;	E	101	65	145	##101;	e
6	6	006	ACK (acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F	102	66	146	##102;	f
7	7	007	BEL (bell)	39	27	047	##39;	'	71	47	107	##71;	G	103	67	147	##103;	g
8	8	010	BS (backspace)	40	28	050	##40;	(72	48	110	##72;	H	104	68	150	##104;	h
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I	105	69	151	##105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J	106	6A	152	##106;	j
11	B	013	VT (vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K	107	6B	153	##107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L	108	6C	154	##108;	l
13	D	015	CR (carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M	109	6D	155	##109;	m
14	E	016	SO (shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N	110	6E	156	##110;	n
15	F	017	SI (shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O	111	6F	157	##111;	o
16	10	020	DLE (data link escape)	48	30	060	##48;	0	80	50	120	##80;	P	112	70	160	##112;	p
17	11	021	DC1 (device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q	113	71	161	##113;	q
18	12	022	DC2 (device control 2)	50	32	062	##50;	2	82	52	122	##82;	R	114	72	162	##114;	r
19	13	023	DC3 (device control 3)	51	33	063	##51;	3	83	53	123	##83;	S	115	73	163	##115;	s
20	14	024	DC4 (device control 4)	52	34	064	##52;	4	84	54	124	##84;	T	116	74	164	##116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U	117	75	165	##117;	u
22	16	026	SYN (synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V	118	76	166	##118;	v
23	17	027	ETB (end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W	119	77	167	##119;	w
24	18	030	CAN (cancel)	56	38	070	##56;	8	88	58	130	##88;	X	120	78	170	##120;	x
25	19	031	EM (end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y	121	79	171	##121;	y
26	1A	032	SUB (substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z	122	7A	172	##122;	z
27	1B	033	ESC (escape)	59	3B	073	##59;	;	91	5B	133	##91;	[123	7B	173	##123;	{
28	1C	034	FS (file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\	124	7C	174	##124;	
29	1D	035	GS (group separator)	61	3D	075	##61;	=	93	5D	135	##93;]	125	7D	175	##125;	}
30	1E	036	RS (record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^	126	7E	176	##126;	~
31	1F	037	US (unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_	127	7F	177	##127;	DEL

Source: www.LookupTables.com

L'accès à cette base de données peut être codé en dur (chemin présent dans le code source) ou configurable lors de l'exécution du programme.

Note : La gestion des fichiers sera vu et décrite dans le cours.

C. Menu principal

Au démarrage de l'application, il est proposé :

- Le mode utilisateur
- Le mode administrateur
- Configuration du chemin du fichier de base de données (facultatif)

Il suffit donc d'entrer une valeur pour déterminer le mode.

D. Le mode Utilisateur

Le mode utilisateur permet de rechercher une chanson et de la lire.

1. Rechercher une chanson

a) Description

L'utilisateur doit rentrer successivement :

- Artiste
- Chanson

Puis, le programme liste les résultats correspondants avec les informations suivantes:

- ID
- Artiste
- Chanson
- Album
- Style

b) Contrainte

La recherche n'est pas exacte, c'est-à-dire que l'utilisateur peut renseigner des champs incomplets voir vide.

Il est donc possible d'avoir plusieurs résultats.

c) Exemples

Voici 3 exemples en utilisant la base de données suivante :

- | | | | |
|----|-----------|---|-------|
| 1. | Marc | / | Test |
| 2. | Jean | / | Poly |
| 3. | Jean-Marc | / | Music |

1. L'utilisateur rentre :

Marc / Test

Réponse :

- | | | | |
|----|------|---|------|
| 1. | Marc | / | Test |
|----|------|---|------|

2. L'utilisateur rentre :

Jean / « » (vide)

Réponse :

- | | | | |
|----|-----------|---|-------|
| 1. | Jean | / | Poly |
| 2. | Jean-Marc | / | Music |

3. L'utilisateur rentre :

Marc / s

Réponse :

- | | | | |
|----|-----------|---|-------|
| 1. | Marc | / | Test |
| 2. | Jean-Marc | / | Music |

Conseil : Utiliser strstr de la library <string.h> pour chercher une chaîne de caractères dans une autre
Attention une fois encore à la casse !

2. Lire une chanson

Pour lire une chanson, l'utilisateur doit renseigner l'ID.

Une fois la chanson démarrée, il a la possibilité :

- Play
- Pause
- Stop
- Exit (retourne au menu principal)

E. Le mode Administrateur

L'administrateur peut :

- Ajouter des chansons dans la base de données
- Supprimer des chansons de la base de données
- Lister les chansons de la base de données

1. Ajouter une chanson

Pour ajouter une chanson, l'administrateur doit renseigner les champs suivants :

- Artiste
- Chanson
- Album (optionnel)
- Style (optionnel)
- Chemin

Une fois les champs remplis, le programme ajoute la ligne à la base de données (ce qui entraîne un réagencement du fichier d'après les règles de la gestion du fichier).

2. Supprimer une chanson

Le mécanisme est similaire à celui de recherche côté Utilisateur :

L'administrateur doit rentrer successivement :

- Artiste
- Chanson

Puis, le programme liste les résultats correspondants avec les informations suivantes:

- ID
- Artiste
- Chanson
- Album
- Style
- Chemin

La recherche n'est pas exacte, c'est-à-dire que l'utilisateur peut renseigner des champs incomplets voir vide.

Il est donc possible d'avoir plusieurs résultats.

Enfin, pour supprimer une ligne, l'administrateur doit renseigner l'ID, le programme supprime donc la ligne à la base de données (ce qui entraîne un réagencement du fichier d'après les règles de la gestion du fichier).

3. Lister les chansons

Le programme affiche l'intégralité de la base de données avec les informations suivantes pour chaque ligne :

- ID
- Artiste
- Chanson
- Album
- Style
- Chemin

4. Retour au menu principal

A la suite d'un ajout ou d'une suppression d'une ligne, ou du listing des chansons, le programme quitte le mode Administrateur et retourne au menu principal.

F. La gestion des erreurs

Attention, il est important de gérer les cas d'erreur de saisie, des accès au fichier, de la library Audio et ne pas bloquer le programme.

Le retour au menu principal doit toujours être possible !

III. Une library audio : FMOD

Quelques infos sur la library FMOD :

- Library Audio
- Open source
- Facile et complète
- Compatible avec le C
- Documentation en ligne :
http://www.fmod.org/documentation/#content/generated/lowlevel_api.html

Exemple de fonctions pouvant être utiles :

- FMOD_System_Create : Créer un objet system audio FMOD
- FMOD_System_Init : Initialise un objet system audio FMOD
- FMOD_System_CreateSound : Charge un mp3
- FMOD_System_PlaySound : Jouer le mp3
- FMOD_System_GetMasterChannelGroup : Récupère le canal son de l'objet système
- FMOD_Channel_GetPaused : Statut du canal (lecture ou pause)
- FMOD_ChannelGroup_SetPaused : Play / Pause du canal
- FMOD_Sound_Release : Stop et libère le mp3
- FMOD_System_Close : Ferme l'objet system audio FMOD
- FMOD_System_Release : Libère l'objet system audio FMOD

IV. Ajouter une interface graphique avec SDL (optionnel)

Quelques infos sur la library SDL :

- Open source
- Facile et complète
- Multimedia
- Compatible avec le C
- Télécharger la lib : <https://www.libsdl.org/>
- Documentation en ligne : <https://wiki.libsdl.org/>

Exemple de fonctions pouvant être utiles :

- General
 - SDL_Init : charge la SDL en mémoire
 - SDL_Quit : libère la SDL de la mémoire
 - SDL_GetError : Retourne une erreur venant de la library
- Affichage
 - SDL_SetVideoMode : Ouvre une fenêtre graphique
 - SDL_WM_SetCaption : Change le titre de la fenêtre
 - SDL_FillRect / SDL_MapRGB : Colorier une surface unie
 - SDL_Flip : Rafraîchir la fenêtre
 - SDL_CreateRGBSurface / SDL_FreeSurface : Allouer / libérer nouvelle surface
 - SDL_BlitSurface : Ajouter une surface
 - SDL_LoadBMP : Charge une image
 - SDL_WM_SetIcon : Associer un icône à la fenêtre
- Evènement
 - SDL_WaitEvent : Attends un évènement
 - SDL_PollEvent : Récupère un évènement
 - SDL_KEYDOWN / SDL_KEYUP : Evènement clavier
 - SDL_MOUSEBUTTONDOWN / SDL_MOUSEBUTTONUP / SDL_MOUSEMOTION : Evènement souris
- Audio
 - SDL_OpenAudio : Initialisation d'une nouvelle couche audio
 - SDL_CloseAudio : Fermeture d'une couche audio
 - SDL_LoadWAV : Charge un fichier WAV
 - SDL_FreeWAV : Libérer le buffer de lecture
 - SDL_MixAudio : Lire le fichier audio
 - SDL_PauseAudio : Suspendre une lecture