

## TD séance n° 10

### Multimédia : Texte

## 5 Exercices

### 5.1 Objets multimédia composites

#### Exercice n°1:

Voyez-vous d'autres exemples d'objets multimédias composites que ceux cités dans le cours ?

Vidéo et texte (pour les sous-titres), Son et texte (fichier de karaoké), ...

#### Exercice n°2:

En quoi l'objet « Karaoké » (fichier .kar) est-il un objet multimédia composite ?

Un fichier de karaoké contient du son pour la musique et le texte des paroles que l'on doit chanter. Donc il inclut deux types de base.

### 5.2 Chaînes de traitements multimédia

#### Exercice n°3:

Pouvez-vous donner un exemple d'une chaîne de traitement multimédia qui permet d'acquérir du son et de restituer du texte ?

Un système de reconnaissance vocal qui permet soit de commander la machine vocalement, soit de dicter du texte. Exemple de logiciel Dragon Naturally Speaking

Où vient alors s'insérer le traitement spécifique qui se rajoute à la chaîne de traitement multimédia entre le Microphone et l'Ecran par exemple ? Quel est-il ?

Le logiciel de traitement s'insère entre le dispositif d'acquisition (avec le format de données associé) et le logiciel de rendu (le traitement de texte). Donc il prend la place du logiciel d'édition dans le cadre du schéma présenté.

#### Exercice n°4:

Pouvez-vous donner un exemple d'une chaîne de traitement multimédia qui permet d'acquérir de l'image et de restituer du texte ?

C'est un logiciel de reconnaissance optique de caractère : on photographie une page, on obtient donc une image et à l'aide d'un logiciel de type OCR (Optical Character Recognition), on interprète les données dans l'image pour en extraire les caractères qui y sont représentés.

Où vient alors s'insérer le traitement spécifique qui se rajoute à la chaîne de traitement multimédia entre le Scanner et l'Ecran par exemple ? Quel est-il ?

C'est aussi le logiciel d'édition qui diffère. Un des logiciels fournissant ce type de fonctionnalité sur le marché est Omnipage.

### 5.3 Bases

#### Exercice n°5:

De la base 10 vers la base 2. Donner l'écriture en base 2 des nombres suivants :

$$A = 10_{(10)} \quad B = 31_{(10)} \quad C = 32_{(10)}$$

$$A = 1010_{(2)} \quad B = 11111_{(2)} \quad C = 100000_{(2)}$$

#### Exercice n°6:

De la base 2 vers la base 10. Convertir en base 10 les nombres suivants :

$$D = 101001_{(2)} \quad E = 10110011_{(2)} \quad F = 1100101_{(2)} \quad G = 100010111_{(2)}$$

## TD séance n° 10

### Multimédia : Texte

$D = 41_{(10)}$   $E = 179_{(10)}$   $F = 101_{(10)}$   $G = 139_{(10)}$

#### Exercice n°7:

Quel est le nombre le plus grand pour chacun de ces couples de nombres ?

H :  $10000_{(2)}$  et  $64_{(10)}$  I :  $10000001_{(2)}$  et  $101_{(10)}$  J :  $10000000_{(2)}$   $128_{(10)}$  K :  $11111111_{(2)}$   $256_{(10)}$

H :  $10000_{(2)} = 16_{(10)} < 64_{(10)}$

I :  $10000001_{(2)} = 129_{(10)} > 101_{(10)}$

J :  $10000000_{(2)} = 128_{(10)} = 128_{(10)}$

K :  $11111111_{(2)} = 255_{(10)} < 256_{(10)}$

## 5.4 Codage du texte

Nous allons voir que de simples fichiers texte, sans gérer de mise en forme particulière à l'exception du retour à la ligne, peuvent déjà être codés différemment.

#### Exercice n°8:

Créez un fichier `texteANSI.txt` sous Windows avec Notepad++ que vous aurez préalablement installé depuis <http://notepad-plus-plus.org/fr/>. Dans ce fichier, mettez le texte « Bonjour ! » et enregistrez-le.

#### Exercice n°9:

Créez alors deux nouveaux fichiers textes `texteUTF8.txt` et `texteUTF16-BE.txt` avec le même texte « Bonjour ! » mais en modifiant le format de codage (cf. menu Encodage, successivement en UTF-8 puis UTF16-BE).

#### Exercice n°10:

Commencez par démarrer un terminal sous Windows (Démarrer / et taper `cmd` dans la zone pour rechercher les programmes). Pour afficher successivement le contenu des deux fichiers dans le terminal utilisez la commande `type` (équivalent de `cat` sous Unix) :

`type nom_de_fichier` (avec successivement les fichiers `texteANSI.txt`, `texteUTF8.txt` puis `texteUTF16-BE.txt`)

Que remarquez-vous ?

L'affichage des fichiers `texteANSI.txt` et `texteUTF8.txt` sont identiques. Par contre l'affiche de `texteUTF16-BE.txt` affiche bien le même texte, mais avec des espaces entre chaque lettre et un caractère « bizarre » (comme un carré) avant le mot Nonjour !

#### Exercice n°11:

En comparant les fichiers `textANSI.txt` `texteUTF8.txt` et `texteUTF16-BE.txt`, à l'aide de Notepad++, que constatez-vous ?

La taille du deuxième fichier (`UTF16-BE.txt`) est un peu plus de deux fois plus gros que le fichier `textANSI.txt`.

En ouvrant les deux fichiers avec le logiciel Bloc-Notes que remarquez-vous ? Comment le logiciel peut connaître l'encodage utilisé par le fichier ?

Quand on ouvre le fichier avec le bloc-note, le logiciel affiche bien le texte tapé. Les premiers octets du fichiers permettent de stocker l'information sur la nature de l'encodage utilisé dans le fichier

#### Exercice n°12:

Voici par exemple une série de caractère codés en ASCII (ANSI) tels que stockés dans un fichier texte.

6C	65	73	20	6E	65	75	6E	65	75	73	20	66	6F	6E	74	20	71	75	65
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

## TD séance n° 10

### Multimédia : Texte

20	63	6C	69	71	75	65	72												

Utilisez le tableau figurant en annexe pour trouver le texte correspondant.

les neuneus font que cliquer

#### Exercice n°13:

En UTF16-BE, le codage sera le suivant :

FE	FF	00	6C	00	65	00	73	00	20	00	6E	00	65	00	75	00	6E	00	65
00	75	00	73	00	20	00	66	00	6F	00	6E	00	74	00	20	00	71	00	75
00	65	00	20	00	63	00	6C	00	69	00	97	00	75	00	65	00	72		

Si l'éditeur de texte le traite comme un fichier avec des caractères codés en ASCII/ANSI, quel sera le texte affiché ?

(Il y a une séparation entre chaque lettre par rapport à la version précédente)

l e s n e u n e u s f o n t q u e c l i q u e r

Cela vous rappelle-t-il quelque chose ?

Le texte est identique (même encodage utilisé en ASCII et en UTF8 pour les caractères dont le code est entre 0 et 255). Comme ici on utilise pas de caractère ayant un vrai code sur 16 bits, le premier octet est toujours à 0. Donc si on interprète les données en ASCII/ANSI, on voit des espace entre chaque lettre (caractère NUL).

## TD séance n° 10

### Multimédia : Texte

### Pour aller plus loin...

#### Chaînes de traitements multimédia

##### Exercice n°1:

Décliner un exemple de chaîne de traitement d'objets multimédias, en vous inspirant en particulier du tableau sur les différents types de fichiers multimédia. Si vous connaissez des logiciels qui permettent de travailler sur une ou plusieurs étapes de la chaîne, sous Linux, sous Windows, mentionnez-les.

Chaîne de traitement du texte

Dispositif d'acquisition : clavier

Encodage de base : UTF-8

Réencodage : UTF-16

Logiciel de traitement : Notepad, gedit, Word, OpenOffice, ...

Logiciel de rendu: idem + cat, ...

Format de sortie: txt, doc, docx; ...

Dispositif de rendu : écran, imprimante, ...

C'est aussi le logiciel d'édition qui diffère. Un des logiciels fournissant ce type de fonctionnalité sur le marché est Omnipage.

##### Exercice n°2:

Vers des domaines inexplorés... ou presque ... Clavier vers Haut-parleurs ? Image vers Vidéo ? ...

Maintenant que vous avez compris, décrivez une chaîne de traitement multimédia de votre invention. Quel traitement spécifique est nécessaire dans cette chaîne de traitement ?

Si l'on conçoit un objet image comme des cases grises (de 0% de noir à 100% de noir) posée sur un damier, et un objet son composé d'une séquence de couples (note, durée), décrivez un algorithme (et oui, c'est ça !) qui permettrait de transformer l'objet image vers un objet son.

Pour toutes les lignes de l'image

Pour toutes les couples de cases d'une ligne

Lire la couleur de la case n

note = Convertir la couleur en un son

Lire la couleur de la case n+1

durée = Convertir la couleur en une durée

Jouer le son(note, durée)

##### Exercice n°3:

Illustrez cette transformation en traitant deux exemples simples :

- Une image toute grise (que des cases à 50% de noir)  
Un seul son qui se qui a toujours la même durée
- Une image « damier » alternant 2 cases noires et 2 cases blanches  
Le son joué sera du type deux tons (le pin-pon des pompiers)

#### Bases

Dans ces exercices, on passera directement d'une base à l'autre sans passer par la base 10. Pour vous faciliter la conversion, penser à regrouper les bits par 4... et oui,  $15_{(10)} = 1111_{(2)} = F_{(16)}$  !

## TD séance n° 10

### Multimédia : Texte

---

#### Exercice n°4: Conversion du binaire vers hexadécimal

Donner l'écriture en base 16 des nombres suivants :

$$V = 101101_{(2)} \quad W = 101101011110_{(2)} \quad X = 100111001110111_{(2)}$$
$$V = 2D_{(16)} \quad W = B5E_{(16)} \quad X = 4E77_{(16)}$$

#### Exercice n°5: Conversion de l'hexadécimal vers le binaire

Donner l'écriture en base 2 des nombres suivants :

$$Y = 24D_{(16)} \quad Z = FE_{(16)}$$
$$Y = 1001001101_{(2)} \quad Z = 11111110_{(2)}$$

## TD séance n° 10

### Multimédia : Texte

#### Annexe

Décimal	Octal	Hex	Binaire	Caractère
0	0	0	00000000	NUL
1	1	1	00000001	SOH
2	2	2	00000010	STX
3	3	3	00000011	ETX
4	4	4	00000100	EOT
5	5	5	00000101	ENQ
6	6	6	00000110	ACK
7	7	7	00000111	BEL
8	10	8	00001000	BS
9	11	9	00001001	HT
10	12	A	00001010	LF
11	13	B	00001011	VT
12	14	C	00001100	FF
13	15	D	00001101	CR
14	16	E	00001110	SO
15	17	F	00001111	SI
16	20	10	00010000	DLE
17	21	11	00010001	DC1
18	22	12	00010010	DC2
19	23	13	00010011	DC3
20	24	14	00010100	DC4
21	25	15	00010101	NAK
22	26	16	00010110	SYN
23	27	17	00010111	ETB
24	30	18	00011000	CAN
25	31	19	00011001	EM
26	32	1A	00011010	SUB
27	33	1B	00011011	ESC
28	34	1C	00011100	FS
29	35	1D	00011101	GS
30	36	1E	00011110	RS
31	37	1F	00011111	US
32	40	20	00100000	SP
33	41	21	00100001	!
34	42	22	00100010	"
35	43	23	00100011	#
36	44	24	00100100	\$
37	45	25	00100101	%
38	46	26	00100110	&
39	47	27	00100111	'
40	50	28	00101000	(

Décimal	Octal	Hex	Binaire	Caractère
41	51	29	00101001	)
42	52	2A	00101010	*
43	53	2B	00101011	+
44	54	2C	00101100	,
45	55	2D	00101101	-
46	56	2E	00101110	.
47	57	2F	00101111	/
48	60	30	00110000	0
49	61	31	00110001	1
50	62	32	00110010	2
51	63	33	00110011	3
52	64	34	00110100	4
53	65	35	00110101	5
54	66	36	00110110	6
55	67	37	00110111	7
56	70	38	00111000	8
57	71	39	00111001	9
58	72	3A	00111010	:
59	73	3B	00111011	;
60	74	3C	00111100	<
61	75	3D	00111101	=
62	76	3E	00111110	>
63	77	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C
68	104	44	01000100	D
69	105	45	01000101	E
70	106	46	01000110	F
71	107	47	01000111	G
72	110	48	01001000	H
73	111	49	01001001	I
74	112	4A	01001010	J
75	113	4B	01001011	K
76	114	4C	01001100	L
77	115	4D	01001101	M
78	116	4E	01001110	N
79	117	4F	01001111	O
80	120	50	01010000	P
81	121	51	01010001	Q

## TD séance n° 10

### Multimédia : Texte

Décimal	Octal	Hex	Binaire	Caractère
82	122	52	01010010	R
83	123	53	01010011	S
84	124	54	01010100	T
85	125	55	01010101	U
86	126	56	01010110	V
87	127	57	01010111	W
88	130	58	01011000	X
89	131	59	01011001	Y
90	132	5A	01011010	Z
91	133	5B	01011011	[
92	134	5C	01011100	
93	135	5D	01011101	]
94	136	5E	01011110	^
95	137	5F	01011111	_
96	140	60	01100000	`
97	141	61	01100001	a
98	142	62	01100010	b
99	143	63	01100011	c
100	144	64	01100100	d
101	145	65	01100101	e
102	146	66	01100110	f
103	147	67	01100111	g
104	150	68	01101000	h
105	151	69	01101001	i
106	152	6A	01101010	j
107	153	6B	01101011	k
108	154	6C	01101100	l
109	155	6D	01101101	m
110	156	6E	01101110	n
111	157	6F	01101111	o
112	160	70	01110000	p
113	161	71	01110001	q
114	162	72	01110010	r
115	163	73	01110011	s
116	164	74	01110100	t
117	165	75	01110101	u
118	166	76	01110110	v
119	167	77	01110111	w
120	170	78	01111000	x
121	171	79	01111001	y
122	172	7A	01111010	z
123	173	7B	01111011	{

Décimal	Octal	Hex	Binaire	Caractère
124	174	7C	01111100	
125	175	7D	01111101	}
126	176	7E	01111110	~
127	177	7F	01111111	DEL