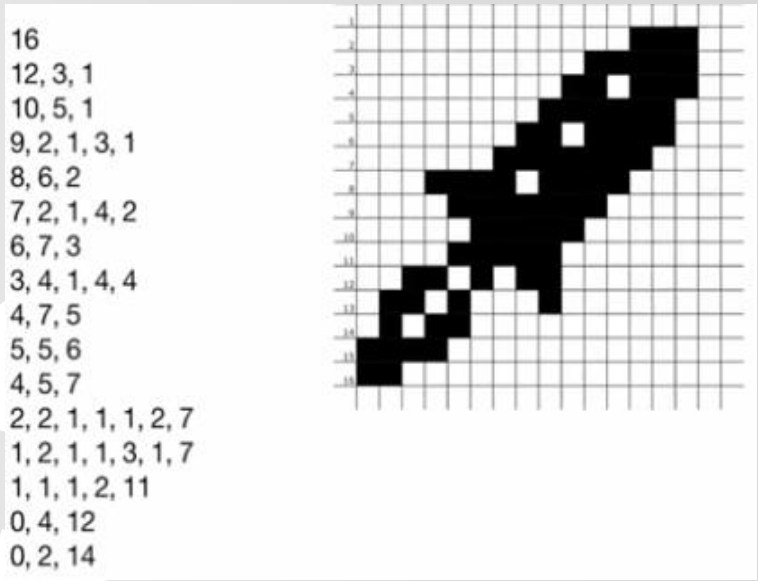
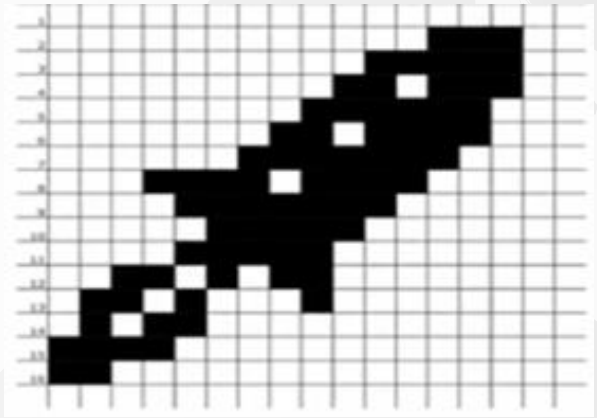
App Inventor basics

Codage et compression des images



Sources : Présentation en grande partie issue de celle du cours mobile CSP de **Ralph Morelli** et al.
Traduction et adaptation françaises par https://onvaessayer.org Pierre Huguet.

L'encodage par plages : run Length (RLE)



```
1111111111111111
1111111111110001
1111111111000001
1111111110010001
1111111100000011
1111111001000011
1111110000000111
1110000100001111
1111000000011111
1111100000111111
1111000001111111
1100101001111111
1001011101111111
1010011111111111
0000111111111111
0011111111111111
```

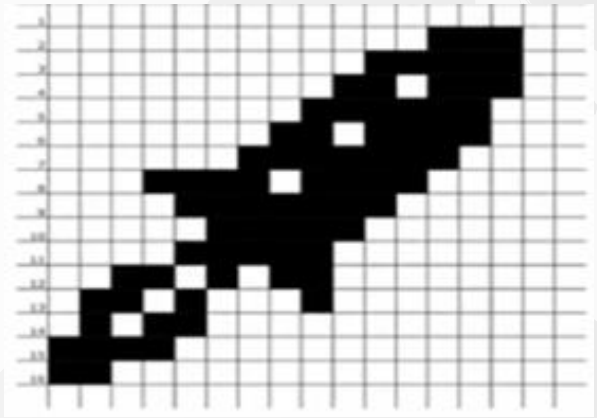
Pixels : 0/1 en monochrome

L'encodage par plages : run Length (RLE)

- La compression RLE consiste à indiquer pour chaque ligne le nombre de pixels blancs, puis noirs, puis blancs, puis noirs, …
  - 16 : la ligne 1 a 16 pixels blancs
  - 12, 3, 1 : la ligne 2, 12 blancs, 3noirs, 1 blanc

16
12, 3, 1
10, 5, 1
9, 2, 1, 3, 1
8, 6, 2
7, 2, 1, 4, 2
6, 7, 3
3, 4, 1, 4, 4
4, 7, 5
5, 5, 6
4, 5, 7
2, 2, 1, 1, 1, 2, 7
1, 2, 1, 1, 3, 1, 7
1, 1, 1, 2, 11
0, 4, 12
0, 2, 14

RLE



```
1111111111111111
1111111111110001
1111111111000001
1111111110010001
1111111100000011
1111111001000011
1111110000000111
1110000100001111
1111000000011111
1111100000111111
1111000001111111
1100101001111111
1001011101111111
1010011111111111
0000111111111111
0011111111111111
```

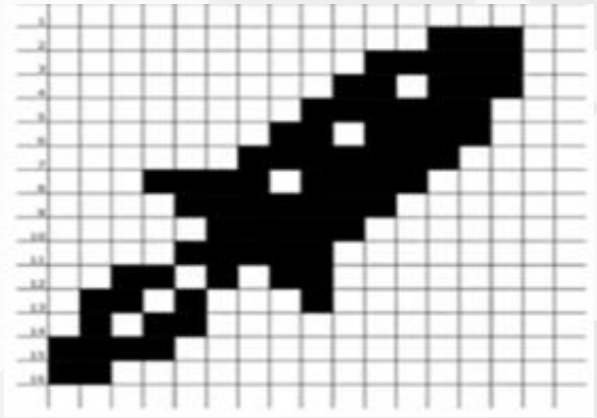
Pixels : 0/1 en monochrome

L'encodage par plages : run Length (RLE)

- RLE est une méthode de compression ***sans perte***.
  adaptée à des images avec beaucoup de blanc
  et utilisée dans le format BMP (***bitmap***)

16
12, 3, 1
10, 5, 1
9, 2, 1, 3, 1
8, 6, 2
7, 2, 1, 4, 2
6, 7, 3
3, 4, 1, 4, 4
4, 7, 5
5, 5, 6
4, 5, 7
2, 2, 1, 1, 1, 2, 7
1, 2, 1, 1, 3, 1, 7
1, 1, 1, 2, 11
0, 4, 12
0, 2, 14

RLE



```
1111111111111111
1111111111110001
1111111111000001
1111111110010001
1111111100000011
1111111001000011
1111110000000111
1110000100001111
1111000000011111
1111100000111111
1111000001111111
1100101001111111
1001011101111111
1010011111111111
0000111111111111
0011111111111111
```

Pixels : 0/1 en monochrome

Compression **avec** ou **sans** perte

- ***Sans perte:*** l'image originale peut être reconstruite à l'identique à partir des données compressées
  - adaptée aux images médicales, ...
  - utilisée dans le format d'image BMP (***bitmap***)
- ***Avec perte:*** des données peuvent être perdues par le processus de compression/décompression
  - Ok pour les photos où l'oeil humain ne peut pas voir la différence,
  - par exemple, le format JPEG
- ***Taux de compression:***
  - rapport entre la taille de l'image initiale et la taille del'image compressée

Réduction pour une image monochrome ?

RLE

```
16
12, 3, 1
10, 5, 1
9, 2, 1, 3, 1
8, 6, 2
7, 2, 1, 4, 2
6, 7, 3
3, 4, 1, 4, 4
4, 7, 5
5, 5, 6
4, 5, 7
2, 2, 1, 1, 1, 2, 7
1, 2, 1, 1, 3, 1, 7
1, 1, 1, 2, 11
0, 4, 12
0, 2, 14
```

RLE

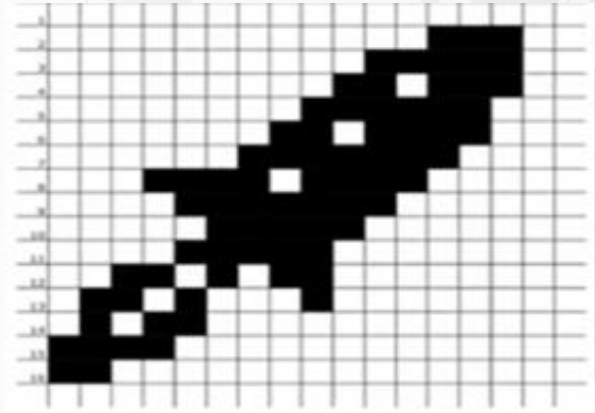


Tableau de pixels

```
1111111111111111
1111111111110001
1111111111000001
1111111110010001
1111111100000011
1111111001000011
1111110000000111
1110000100001111
1111000000011111
1111100000111111
1111000001111111
1100101001111111
1001011101111111
1010011111111111
0000111111111111
0011111111111111
```

Pixels : 0/1 en monochrome

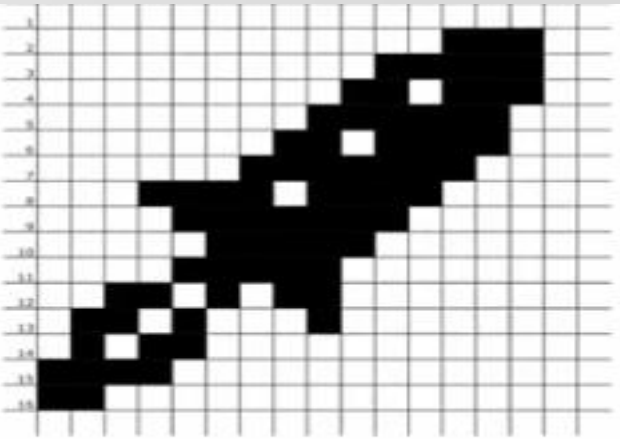
62 nombres
~8 bits/nombre
496 bits

>

16 x 16 = 256 pixels
1 bit/pixel
256 bits

Image en 256 couleurs : 8 bit/pixel

- Technique des années 1980…
- Rouge, Vert, Bleu (RVB)
- 8 bits par pixel
- 3 R, 3 V, 2 B



0	1	2	3	4	5	6	7
R	R	R	V	V	V	B	B
1	1	1	0	0	0	1	1

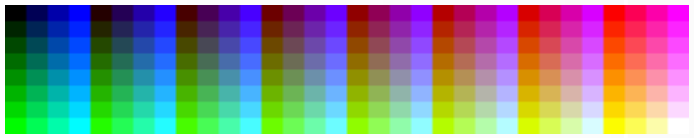
- 0000 0000 = noir(0)
- 1111 1111 = blanc (255)

Marie-Caroline Gries, Pierre Huguet

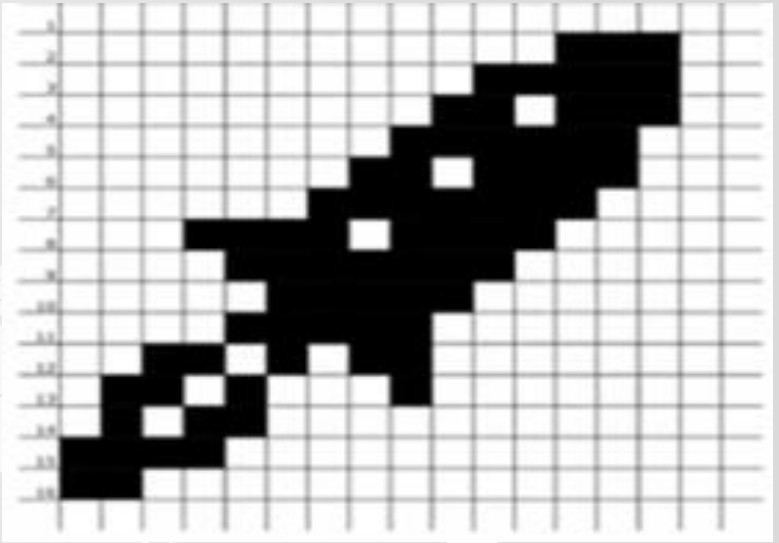
RLE modifié : 256 couleurs

Si il y a plus de 2 couleurs, il faut 2 nombres par plage :

- un pour la couleur
- un pour la longueur de la plage



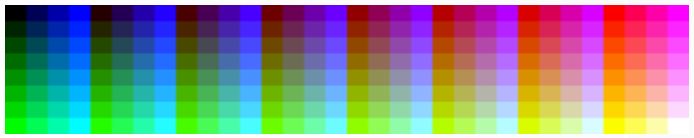
255,16
255,12, 0,3 255,1
255,10, 0,5, 255,1
255,9, 0,2, 255,1, 0,3, 255,1
255,8, 0,6, 255,2
255,7, 0,2, 255,1, 0,4, 255,2
255,6, 0,7, 255,3
255,3, 0,5, 255,1, 0,4, 255,4
255,4, 0,7, 255,5
255,5, 0,5, 255,6
255,4, 0,5, 255,7
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7
255,1, 0,1, 255,1, 0, 2, 255, 11
255,0, 0,4, 255,12
255,0, 0,2, 255,14



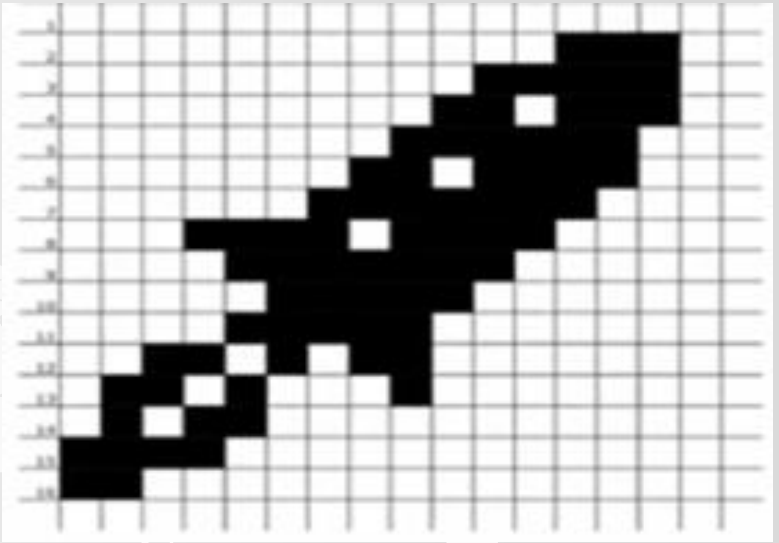
RLE modifié : 256 couleurs

Si il y a plus de 2 couleurs, il faut 2 nombres par plage :

- un pour la couleur
- un pour la longueur de la plage



255,16
255,12, 0,3 255,1
255,10, 0,5, 255,1
255,9, 0,2, 255,1, 0,3, 255,1
255,8, 0,6, 255,2
255,7, 0,2, 255,1, 0,4, 255,2
255,6, 0,7, 255,3
255,3, 0,5, 255,1, 0,4, 255,4
255,4, 0,7, 255,5
255,5, 0,5, 255,6
255,4, 0,5, 255,7
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7
255,1, 0,1, 255,1, 0, 2, 255, 11
255,0, 0,4, 255,12
255,0, 0,2, 255,14



124 nombres x 8 bits/nombre

992 bits

<

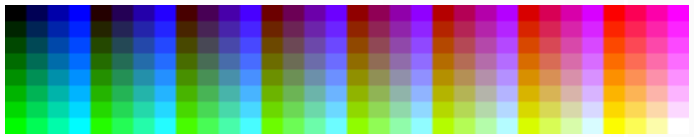
256 pixels x 8 bits/pixel

2048 bits

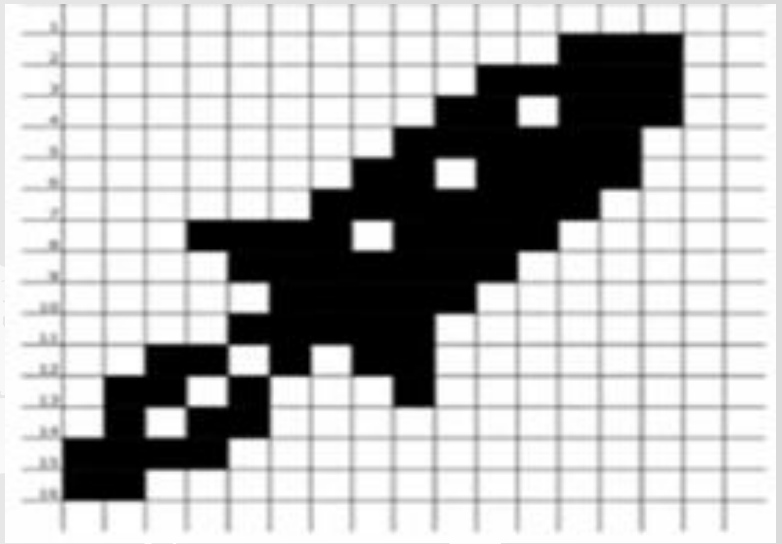
RLE modifié : 256 couleurs

Si il y a plus de 2 couleurs, il faut 2 nombres par plage :

- un pour la couleur
- un pour la longueur de la plage



255,16
255,12, 0,3 255,1
255,10, 0,5, 255,1
255,9, 0,2, 255,1, 0,3, 255,1
255,8, 0,6, 255,2
255,7, 0,2, 255,1, 0,4, 255,2
255,6, 0,7, 255,3
255,3, 0,5, 255,1, 0,4, 255,4
255,4, 0,7, 255,5
255,5, 0,5, 255,6
255,4, 0,5, 255,7
255,2, 0,2, 255,1, 0,1, 255,1, 0,2, 255,7
255,1, 0,2, 255,1, 0,1, 255,3, 0,1 255,7
255,1, 0,1, 255,1, 0, 2, 255, 11
255,0, 0,4, 255,12
255,0, 0,2, 255,14



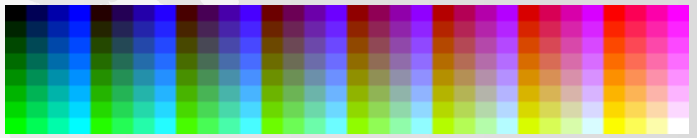
Taux de compression :
2048/992 = 2,1

124 nombres x 8 bits/nombre
992 bits

<

256 pixels x 8 bits/pixel
2048 bits

16 millions de couleurs sur 24 bits



- Rouge, Vert, Bleu (RVB)
- 24 bits par pixel
  - 8 R, 8 V, 8 B
- 2^{24} = 16,777,216 couleurs
  c'est plus q'on ne peut voir

Pour la couleur : 3 nombres par pixel

- 0, 0, 0 = noir
- 255,255,255 = blanc
- 255, 0, 200 =

16 millions de couleurs sur 24 bits

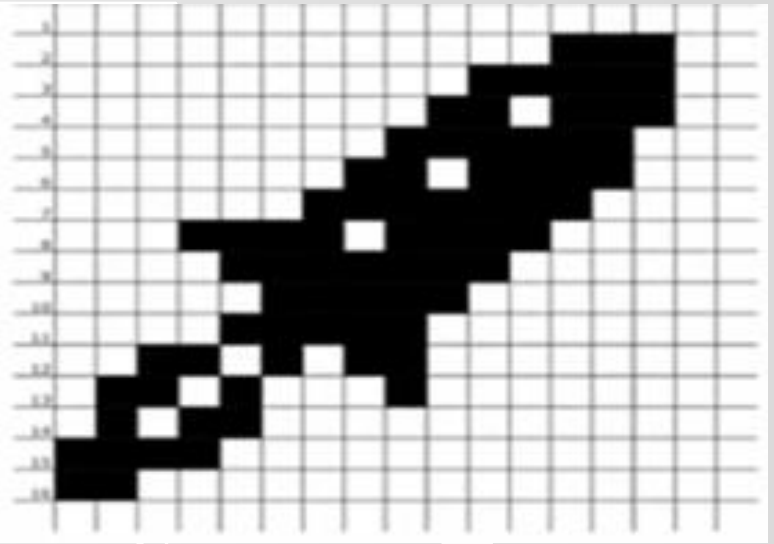
Avec 16 millions de couleurs, il faut 4 nombres par plage :

- Trois pour la couleur
- un pour la longueur de la plage

255,255,255,16
255,255,255,12, 0,0,0,3 255,255,255,1

...

255,255,255,0, 0,0,0,4, 255,255,255,12
255,255,255,0, 0,0,0,2, 255,255,255,14



Taux de compression :
6144/1984 = 3,1

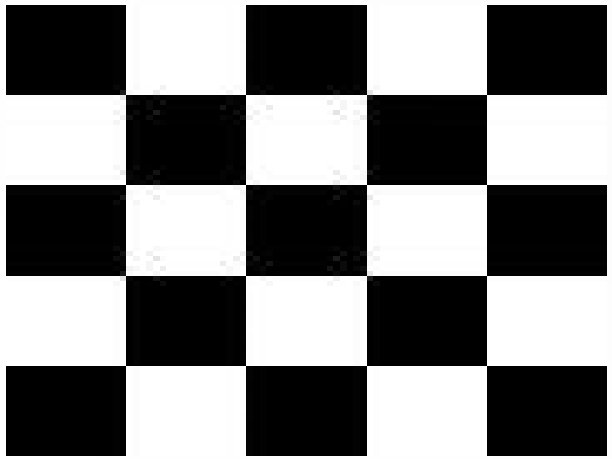
248 nombres avec 8 bits/nombre
1984 bits

<

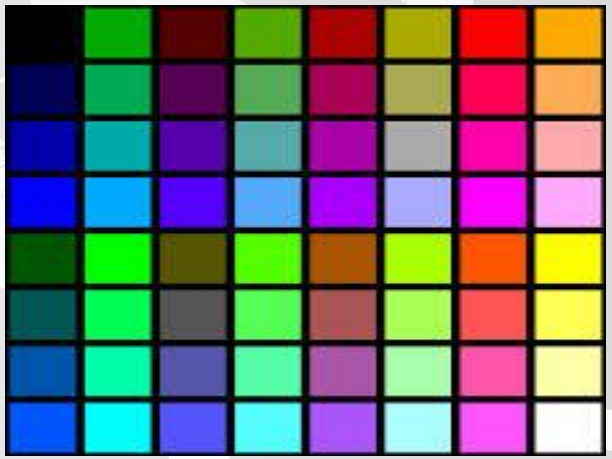
256 pixels x 24 bits/pixel
6,144 bits

Quel taux de compression RLE?

- Dépends du nombre de bits/pixel
- Mais, dépend encore plus...
- ...*du nombre de couleurs différentes dans l'image*



Plus il y a de couleurs,
plus elles changent,
donc plus il faut de
nombres.



Le système binaire permet de

- représenter des nombres
- représenter des couleurs
- représenter des caractères

Représentation des caractères

- Le codage ASCII des caractères (1960…)
  American Standard Code for Information
- **ASCII table**

Hex	Symbol	Hex	Symbol	Hex	Symbol	Hex	Symbol	Hex	Symbol	Hex	Symbol
20		30	0	40	@	50	P	60	`	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	

Représentation des caractères

- ASCII : quelques exemples

Binary	Hex	Decimal	Character
0010 0000	20	32	Space
0011 0000	30	48	0
0100 0001	41	65	A
0110 0001	61	97	a

Donc,...

Quel est le caractère correspondant à cette séquence de bits ?

0100 0001

Est-ce que c'est

le nombre 65? ou la lettre 'A' ou la couleur

Ca dépend du contexte !

Si 0100 0001

est utilisé pour

pour une Image ?

A

pour un Texte ?

65

pour un nombre ?

ou le langage machine ?