

# Ensimag 1A - Projet Logiciel en C

## Décodeur JPEG

François Broquedis, Matthieu Chabanas,  
Julie Dumas, Julien Maillard, Olivier Muller

2 mai 2023

# Décodeur JPEG

```
ff d8 ff e0 00 10 4a 46 49
46 00 01 01 01 00 60 00 60
00 00 ff fe 00 12 53 68 61
75 6e 20 74 68 65 20 73 68
65 65 70 00 ff db 00 43 00
0a 07 07 09 07 06 0a 09 08
09 0b 0b 0a 0c 0f 19 10 0f
0e 0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28 2d
39 30 28 2a 36 2b 22 23 32
44 32 36 3b 3d ...
```

# Décodeur JPEG

```
ff d8 ff e0 00 10 4a 46 49
46 00 01 01 01 00 60 00 60
00 00 ff fe 00 12 53 68 61
75 6e 20 74 68 65 20 73 68
65 65 70 00 ff db 00 43 00
0a 07 07 09 07 06 0a 09 08
09 0b 0b 0a 0c 0f 19 10 0f
0e 0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28 2d
39 30 28 2a 36 2b 22 23 32
44 32 36 3b 3d ...
```



# Décodeur JPEG

```
ff d8 ff e0 00 10 4a 46 49
46 00 01 01 01 00 60 00 60
00 00 ff fe 00 12 53 68 61
75 6e 20 74 68 65 20 73 68
65 65 70 00 ff db 00 43 00
0a 07 07 09 07 06 0a 09 08
09 0b 0b 0a 0c 0f 19 10 0f
0e 0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28 2d
39 30 28 2a 36 2b 22 23 32
44 32 36 3b 3d ...
```



3 semaines

# Décodeur JPEG

```
ff d8 ff e0 00 10 4a 46 49
46 00 01 01 01 00 60 00 60
00 00 ff fe 00 12 53 68 61
75 6e 20 74 68 65 20 73 68
65 65 70 00 ff db 00 43 00
0a 07 07 09 07 06 0a 09 08
09 0b 0b 0a 0c 0f 19 10 0f
0e 0e 0f 1e 16 17 12 19 24
20 26 25 23 20 23 22 28 2d
39 30 28 2a 36 2b 22 23 32
44 32 36 3b 3d ...
```

image JPEG 14 Ko



image PPM 202 Ko

# Sommaire

- 1 Introduction & organisation
- 2 Bon, mais c'est quoi le JPEG ?
- 3 Travail attendu

# Objectifs du Projet Logiciel en C

## Apprentissage d'un langage de référence : le C

### Mise en pratique des connaissances acquises dans d'autres modules

- Algo/Prog, CEP, Théorie de l'information, etc.
- Outils de développement : gdb/ddd, valgrind, gprof, etc.

### Premier aperçu des contraintes d'un « gros » projet logiciel

- Travail en équipe : répartition des tâches, mise en commun
- Gestion du temps et de la difficulté des différentes étapes
- Respect d'un cahier des charges
- Importance de la phase de validation (tests, debug, ...)
- Utilisation d'un gestionnaire de gestion (git) pour de vrai !

# Compétences travaillées

- **Compréhension** du format (notions de TI, d'algo), lecture d'une vraie doc (JPEG )
- Du **C varié** : mémoire, tableaux, structures/fonctions, bas niveau, ... des pointeurs aussi !
- Travail de **conception**, structuration (génie logiciel)
- Travail en **équipe**
- Plein de trucs à faire, **à vous de vous organiser** !



# Vos profs des 3 prochaines semaines



Julie Dumas



Julien Maillard



François Broquedis & Olivier Muller



Matthieu Chabanas

# A quoi servent ces profs ?

## Les encadrants et encadrantes sont là pour

- vous aider sur la programmation en C en général
- vous aider sur les points techniques du JPEG
- vous conseiller sur comment structurer son projet
- vous conseiller sur comment s'organiser

## Les encadrants et encadrantes NE SONT PAS LÀ pour

- faire le projet à votre place
- vous donner les solutions à tous les problèmes
- remplacer les outils de debug
- assurer une hotline 24/7

# Planning / Encadrement

Libre-service encadré : du **mardi 2 mai** au **mardi 23 mai**

- 24 demi-journées d'encadrement
- Au moins 1 prof chaque demi journée, sauf we et jours fériés
- Salles machines du 1<sup>er</sup> étage

Readme & point d'étape

- Fichier Readme.md à jour dans votre dépôt git, schémas
- RDV avec un enseignant ou une enseignante, d'ici 3 à 8 jours

Date du rendu : **mercredi 24 mai 12h00**

- Pas d'encadrement le mercredi matin !

Soutenances

- **jeudi 25 mai** et **vendredi 26 mai**

# Evaluation

Projet C : 2/3 de la note finale du cours *Projet Logiciel en C*

(Nb : non rattrapable ; 50% de la note de session 2)

Soutenance de 40 minutes

- 10-15 minutes de démonstration
- questions et tests de l'examineur ou l'examinatrice
- tout le trinôme est présent
- sur les machines de l'Ensimag
- le code évalué sera celui rendu le mercredi midi

# Evaluation

Projet C : 2/3 de la note finale du cours *Projet Logiciel en C*

(Nb : non rattrapable ; 50% de la note de session 2)

Soutenance de 40 minutes

- 10-15 minutes de démonstration
- questions et tests de l'examineur ou l'examinatrice
- tout le trinôme est présent
- sur les machines de l'Ensimag
- le code évalué sera celui rendu le mercredi midi
- **dress to impress ?**

# Comment réussir son projet ?

# Comment réussir son projet ?

Plusieurs solutions :

- ① **Je bosse**
- ② **J'achète mes profs**
- ③ **Je triche**

# C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparait pas clairement (par exemple en commentaires) est considéré comme frauduleux.**



## C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparaît pas clairement (par exemple en commentaires) est considéré comme frauduleux.**

Fraude ou pas fraude ?

## C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparaît pas clairement (par exemple en commentaires) est considéré comme frauduleux.**

Fraude ou pas fraude ?

- *J'ai pris un bout de code sur internet qui calcule une iDCT rapide ;*
- *Idem, mais j'ai indiqué la référence et je le dis à la soutenance ;*

## C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparaît pas clairement (par exemple en commentaires) est considéré comme frauduleux.**

### Fraude ou pas fraude ?

- *J'ai pris un bout de code sur internet qui calcule une iDCT rapide ;*
- *Idem, mais j'ai indiqué la référence et je le dis à la soutenance ;*
- *J'ai bossé avec un membre d'un autre trinôme sur le module machin, au final on a le même code ;*

## C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparaît pas clairement (par exemple en commentaires) est considéré comme frauduleux.**

### Fraude ou pas fraude ?

- *J'ai pris un bout de code sur internet qui calcule une iDCT rapide ;*
- *Idem, mais j'ai indiqué la référence et je le dis à la soutenance ;*
- *J'ai bossé avec un membre d'un autre trinôme sur le module machin, au final on a le même code ;*
- *J'ai croisé un vieil Ensimag promo 1789 qui avait fait le même projet, il m'a filé des bouts de code ;*

# C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparaît pas clairement (par exemple en commentaires) est considéré comme frauduleux.**

## Fraude ou pas fraude ?

- *J'ai pris un bout de code sur internet qui calcule une iDCT rapide ;*
- *Idem, mais j'ai indiqué la référence et je le dis à la soutenance ;*
- *J'ai bossé avec un membre d'un autre trinôme sur le module machin, au final on a le même code ;*
- *J'ai croisé un vieil Ensimag promo 1789 qui avait fait le même projet, il m'a filé des bouts de code ;*
- *Mieux ! J'ai trouvé la page github d'un ancien qui fait le malin avec son portfolio de projets Ensimag, et avec son upsampler mon projet fonctionne !*

# C'est quoi la fraude, pour nous ?

**Tout code qui n'a pas été écrit par vous, et dont la source n'apparaît pas clairement (par exemple en commentaires) est considéré comme frauduleux.**

## Fraude ou pas fraude ?

- *J'ai pris un bout de code sur internet qui calcule une iDCT rapide ;*
- *Idem, mais j'ai indiqué la référence et je le dis à la soutenance ;*
- *J'ai bossé avec un membre d'un autre trinôme sur le module machin, au final on a le même code ;*
- *J'ai croisé un vieil Ensimag promo 1789 qui avait fait le même projet, il m'a filé des bouts de code ;*
- *Mieux ! J'ai trouvé la page github d'un ancien qui fait le malin avec son portfolio de projets Ensimag, et avec son upsampler mon projet fonctionne !*
- *J'ai débloqué un copain en galère en lui ré-expliquant le codage RLE.*

# Et il m'arrive quoi si je fraude ?

De **0 au projet**, le **refaire**, jusqu'à un **passage en commission de discipline de l'INP**.

## Rappels

- Note de Projet Logiciel en C =  $1/3 * \text{prépa} + 2/3 * \text{projet}$
- UE *Algorithmique et Langage C* :
  - Algo et structures de données : coeff 3 ;
  - Projet Logiciel en C : coeff 4.

Mais si je suis pas pris, je risque rien !



FIGURE – Un ancien élève anonyme dont le projet fonctionne après un chouette copier/coller sur internet



# Mais si je suis pas pris, je risque rien !

```

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/
>>>> file: LongJump.py
#S

print("***** Long Jump Information System *****")
print("Please enter the names of competitors. (Press return when done)")
print("Competitor no. 1:")
competitor = input()
b,c,g,h,d,k = 1,0,0,0,[],0
maxi,competitors = [],[competitor]
while True:
    b += 1
    print("Competitor no. "+str(b)+":")
    competitor = input()
    if competitor == "":break
    else:
        competitors.append(competitor)
print("Please enter the distances for each competitor.")
for each in competitors:
    print("Competitor " + each + " sep="")
    at1 = input("Attempt 1:\n")
    at2 = input("Attempt 2:\n")
    at3 = input("Attempt 3:\n")
    x = (at1+at2+at3).lower()
    if (at1+at2+at3).find("oul") != -1:
        x = (at1+at2+at3).lower()
    d.append(at1)
    d.append(at2)
    d.append(at3)
    x = (at1+at2+at3).lower()
    maxi.append(max(eval(at1),eval(at2),eval(at3)))

```

```

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/
>>>> file: LongJump.py

print("***** Long Jump Information System *****")
print("Please enter the names of competitors. (Press return when done)")
print("Competitor no. 1:")
competitor = input()
b,c,g,h,d,k = 1,0,0,0,[],0
maximums,competitors = [],[competitor]
while True:
    b += 1
    print("Competitor no. "+str(b)+":")
    competitor = input()
    if competitor == "":break
    else:
        competitors.append(competitor)
print("Please enter the distances for each competitor.")
for each in competitors:
    print("Competitor " + each + " sep="")
    attempt1 = input("Attempt 1:\n")
    attempt2 = input("Attempt 2:\n")
    attempt3 = input("Attempt 3:\n")
    g = (attempt1+attempt2+attempt3).lower()
    if (attempt1+attempt2+attempt3).find("oul") != -1:
        g = (attempt1+attempt2+attempt3).lower()
    d.append(attempt1)
    d.append(attempt2)
    d.append(attempt3)
    if (attempt1+attempt2+attempt3).find("oul") != -1:
        maximums.append(max(eval(attempt1),eval(attempt2),eval(attempt3)))
    else:
        d.remove("foul")

```

FIGURE – Illustration de l'utilisation de MOSS sur deux projets python.

# Mais si je suis pas pris, je risque rien !

Donc pourquoi pas, mais...

- Si vous fraudez, faites le bien !
- Ne vous faites pas prendre
- Ca ne vous prendra pas moins de temps que de faire le projet
- Intérêt ?

# Sommaire

- 1 Introduction & organisation
- 2 Bon, mais c'est quoi le JPEG ?
- 3 Travail attendu

# JPEG ?

- *Joint Photographic Experts Group* : comité de normalisation pour la compression d'images
- A l'origine de la norme ISO/IEC IS 10918-1 | ITU-T Recommendation T.81, plus communément appelé JPEG d'après le nom du comité
- Norme complexe, nombreux algorithmes de compression
- Plusieurs formats d'échange supportent le JPEG, le JFIF (JPEG File Interchange Format) est le plus courant

# Avantages

- Format incoutournable pour le stockage d'image, malgré son âge respectable :  
début des années 90... encore dans vos smartphones aujourd'hui !
- Puissance requise très faible pour l'encodage/décodage  
→ pratique à utiliser dans les appareils photos numériques par exemple, internet, ...
- Très bon rapport qualité / taux de compression

# Principe du codec JPEG

## Image brute non compressée

- $n \times m$  pixels en RGB  $\rightarrow$  partitionnée en blocs  $8 \times 8$



## Compression des données de l'image

# Principe du codec JPEG

## Image brute non compressée

- $n \times m$  pixels en RGB  $\rightarrow$  partitionnée en blocs  $8 \times 8$

## Compression des données de l'image

- Changement de représentation : RGB  $\rightarrow$  YCbCr
- Sous échantillonnage de certaines composantes

```

70 74 6d 99 c8 b7 bd c3
73 74 8b c9 b9 d6 8f 83
6c 9f cf c3 63 8f bc 99
6f 62 c0 ab 6b c3 cb c9
72 7c ce b8 bc c4 c1 c4
73 4e 72 a1 bd c6 c6 c2
71 90 69 88 be c4 c3 c3
89 a2 21 91 8d be c6 ab

```

# Principe du codec JPEG

## Image brute non compressée

- $n \times m$  pixels en RGB  $\rightarrow$  partitionnée en blocs  $8 \times 8$

## Compression des données de l'image

- Changement de représentation : RGB  $\rightarrow$  YCbCr
- Sous échantillonnage de certaines composantes
- Changement de domaine : spatial  $\rightarrow$  fréquentiel (DCT)

```
00f2 ff36 ffca 0008 fff2 001f 0000 fdd
0005 0028 ffc b fcd fff6 0017 0023 000b
ffda fff5 0015 005e 0027 ffca fbb fdf
0018 ffc2 ffe0 0023 0031 ffde 0010 0030
000c ffe8 0022 0000 0000 0000 0000 0000
fffc ffe4 002c ffe6 fff1 0000 0000 0011
ffed 0000 001f ffe f 0000 0017 ffe8 0000
001b 0023 0000 ffd9 ffeb ffed ffeb 0013
```



# Principe du codec JPEG

## Image brute non compressée

- $n \times m$  pixels en RGB  $\rightarrow$  partitionnée en blocs  $8 \times 8$

## Compression des données de l'image

- Changement de représentation : RGB  $\rightarrow$  YCbCr
- Sous échantillonnage de certaines composantes
- Changement de domaine : spatial  $\rightarrow$  fréquentiel (DCT)
- Quantification des valeurs, réorganisation spatiale (zig-zag)

```
0030 ffb0 0001 fff9 0005 fffc 0000 fffe
fffe 0006 0003 fff6 0002 fffd 0000 0001
fffe 0017 fffa fffd 0000 ffff ffff 0002
0008 0007 0003 0000 fffe 0001 fffe 0002
0000 0006 0000 0001 0001 0000 0000 0000
fffc fffa 0000 ffff 0000 0000 0000 0000
0000 fffe 0000 0000 0000 0001 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000
```

# Principe du codec JPEG

## Image brute non compressée

- $n \times m$  pixels en RGB  $\rightarrow$  partitionnée en blocs  $8 \times 8$

## Compression des données de l'image

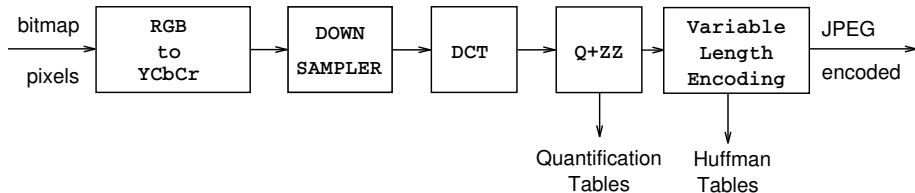
- Changement de représentation : RGB  $\rightarrow$  YCbCr
- Sous échantillonnage de certaines composantes
- Changement de domaine : spatial  $\rightarrow$  fréquentiel (DCT)
- Quantification des valeurs, réorganisation spatiale (zig-zag)

## Stockage efficace de l'information

- Codage DPCM (différences), Codage RLE (répétition de 0)
- Codage de Huffman
- Flux bit à bit
- Format de fichier JFIF : marqueurs, données

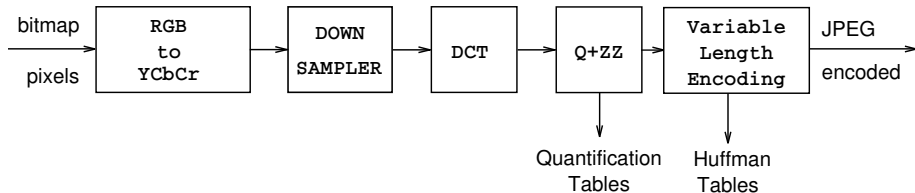
# Principe du codec JPEG

## Encodage

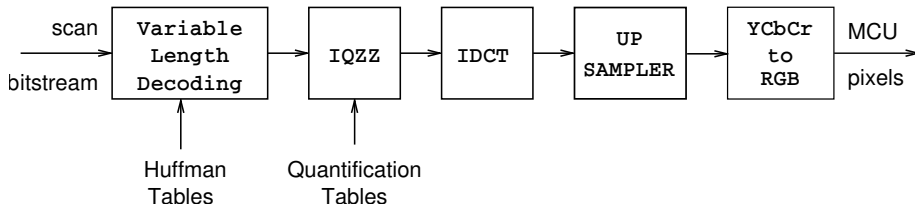


# Principe du codec JPEG

## Encodage



## Décodage



# Compression JPEG

## mode *baseline sequential*

- format le plus simple
- JFIF, DCT, Huffman, précision 8 bits

## mode *progressive* (et autres gourmandises)



# Sommaire

- 1 Introduction & organisation
- 2 Bon, mais c'est quoi le JPEG ?
- 3 Travail attendu**

# Objectifs techniques de ce projet

## Décodeur JPEG (baseline)

- à réaliser intégralement, en C, from scratch
- entrée : des vraies images JPEG, compressées
- sortie : les mêmes mais vachement plus grosses, format brut PPM
- jpeg2ppm shaun\_the\_sheep.jpeg

## Extension : mode progressif

## Critères d'évaluation

- **Fonctionnel**
- **Qualité** (conception, factorisation, efficacité, lisibilité, évolutif, ...)

# Vous avez à votre disposition (mode *hardcode*)



INTERNATIONAL TELECOMMUNICATION UNION

## CCITT

THE INTERNATIONAL  
TELEGRAPH AND TELEPHONE  
CONSULTATIVE COMMITTEE

## T.81

(09/92)

**TERMINAL EQUIPMENT AND PROTOCOLS  
FOR TELEMATIC SERVICES**

---

**INFORMATION TECHNOLOGY –  
DIGITAL COMPRESSION AND CODING  
OF CONTINUOUS-TONE STILL IMAGES –  
REQUIREMENTS AND GUIDELINES**



# Vous avez à votre disposition

- un **sujet** qui explique tout bien : les étapes de l'encodage, le format, des exemples et tout
- de **belles images** pour tester
- des **outils** pour vous aider à comprendre et valider
- des **conseils** sur comment aborder ce projet
- mais... **0 ligne de C**

# Vous avez à votre disposition

- un **sujet** qui explique tout bien : les étapes de l'encodage, le format, des exemples et tout
- de **belles images** pour tester
- des **outils** pour vous aider à comprendre et valider
- des **conseils** sur comment aborder ce projet
- mais... **0 ligne de C**

- **UN indice** : par où commencer ?

entête (paramètres, tables)	données compressées
-----------------------------	---------------------

# Vous avez à votre disposition

- un **sujet** qui explique tout bien : les étapes de l'encodage, le format, des exemples et tout
- de **belles images** pour tester
- des **outils** pour vous aider à comprendre et valider
- des **conseils** sur comment aborder ce projet
- mais... **0 ligne de C**

- **UN indice** : par où commencer ? **par le début !**

entête (paramètres, tables)	données compressées
-----------------------------	---------------------

# Outils à disposition

## jpeg2blabla

- décodeur fonctionnel avec traces
- paramètres des images, tables de quantification et Huffman
- valeurs numériques à chaque étape du décodage :  
bloc, iquant, izz, idct, *upsampling*, etc.

## Autre

- hexdump : visualisation binaire
- gimp, cjpeg : tests supplémentaires ?
- et bien sûr les indispensables : gdb, ddd, valgrind, git, ...

# Approche incrémentale

## Décoder des images de complexité croissante

- niveaux de gris, taille  $8 \times 8$
- niveaux de gris, taille quelconque
- couleurs, sans sous-échantillonnage
- couleurs, avec sous-échantillonnage
- ...

## Approche incrémentale

- décodeur **fonctionnel** le plus rapidement possible, cas simples
- gestion d'images de plus en plus complexes
- optimisations

# Quelques conseils

## Lire le sujet

Non mais, sérieux, **lisez le sujet**. Tout est dedans. TOUT. Normalement.

## Conception de votre architecture logicielle

- définir vos modules, fonctions, types de données, ...
- PRIMORDIAL, et à faire **en équipe** avant d'attaquer le développement à proprement parler
- et il faudra REPRENDRE cette conception, plusieurs fois. Eh oui, et c'est normal.

# Quelques conseils (suite)

## Programmation

- programmation incrémentale & itérative
- testez, testez, testez tout le temps. (Tester, aleter, protéger?)
- versionnez, aussi (git)
- **La qualité prime sur la quantité !** Mieux vaut un code lisible, bien conçu et testé que plus de choses non validées...

# Quelques conseils (fin)

## De manière générale

- Ce n'est pas encore les vacances, ne pas traîner pour s'y mettre !
- Difficile... mais gratifiant ?
- **Happiness only real when shared** ou **L'enfer, c'est les autres ?**
- RH : l'autre c'est plus mon copain / ma copine... venez **VITE** nous raconter ça (pas à la soutenance !)



# Suivi & point d'étape

## Sur votre dépôt git

- fichier Readme.md, à tenir à jour *régulièrement*
- un schéma vos mille mots ?

## Contenu

- découpage des fonctionnalités en étapes, modules avec I/O
- architecture C : structures de données, fonctions, ...
- répartition des tâches (.txt, Trello, Gantt chart...)

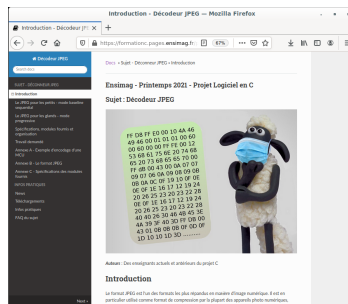
## Interview with the Vampire

- basée sur les éléments ci-dessus
- quand vous êtes prêtes et prêts, mais entre le vendredi 5 mai et le milieu/fin de semaine suivante

# Let's rock !

## Un seul point d'entrée pour toutes les ressources JPEG

<https://formationc.pages.ensimag.fr/projet/jpeg/jpeg/>



## Bon courage... et enjoy !