

# Guide de rédaction de rapport

**Ecole Sup Galilée - Coursus Ingénieur**

**John Chaussard**

LAGA – Université Paris 13

[chaussard@math.univ-paris13.fr](mailto:chaussard@math.univ-paris13.fr)



# Comment rédiger un rapport (de manière générale) ?

Un rapport de TP ou de projet doit montrer une compréhension du sujet.

- Vous devez démontrer que vous avez compris **l'objectif pédagogique du travail**, et **répondant de façon précise et approfondie** aux questions.
- Ne répondez pas « en surface » aux questions, mais allez au-delà en essayant de comprendre ce qui était souhaité par l'enseignant, et en lui montrant que vous l'avez compris.
- Le rapport doit être une sorte de journal de bord : n'hésitez pas à y expliquer vos échecs ainsi que comment vous les avez surmontés.
- Vous devez être capables, des années plus tard, de reprendre ce rapport et comprendre ce qui était demandé, ce que vous avez fait et comment vous y êtes parvenus.



# Présentation du rapport

Ne négligez pas la présentation de votre rapport, car elle donnera une première impression importante au correcteur.

- Une couverture, avec vos noms, la date, le nom de l'école et de l'enseignant, le nom de la matière, le numéro de TP ou le nom du projet, et une petite image pour agrémenter le tout.
- Une page blanche, puis une table des matières (n'oubliez pas de numéroté vos pages).
- Un découpage en section/sous-section/paragraphe, afin de rendre la lecture plus aisée.



# Présentation du rapport

## Devoir de modélisation - seconde partie

### I/ Systeme de vente de vote: Mise à jour

#### 1/ Diagramme de cas d'utilisation

Il faut que le diagramme de cas d'utilisation incorpore la vente et l'achat de votes.

Pour cela j'ai ajouté plusieurs cas d'utilisation(en jaune sur le diagramme) pour pouvoir consulter, acheter et vendre des votes.

De plus il y a maintenant un acteur(secondaire) banque qui permet d'autoriser la vente et de faire la transaction.

Exemple (à ne pas suivre) de couverture d'un rapport reçu par le passé...



# Présentation du rapport

Le style d'écriture doit être soigné, afin de rendre la lecture plus fluide et rapide. Plus le correcteur va vite sur votre rapport, moins il s'attardera sur vos erreurs...

- . Ne choisissez pas une police de caractères trop grande pour gagner de la place, ça se voit.
- . Conservez le même style de police à travers tout le rapport : ne modifiez pas d'un seul coup la taille ou la police utilisée.
- . Evitez des sauts de page trop importants.
- . Utilisez un style différent pour les titres de vos sections, afin qu'ils ressortent bien. Cependant, évitez d'utiliser de la couleur car elle risque de se perdre à l'impression.



# Présentation du rapport

## **Attribut de marché**

- **bullVendre** : Cet attribut est un tableau d'objet "Bulletin", qui représente les bulletins en ventes sur le marché.

## **Méthodes de marché**

- **ajouterOffre** : Permet d'ajouter un bulletin au marché.

- **supprimerOffre** : Permet de supprimer un bulletin. Cette fonction est utilisée si un bulletin a été acheté, ou si le bulletin n'a pas trouvé d'acheteur.

- **prelevement** : Permet de ponctionner 1 % lors d'une transaction, puis de le verser à la banque de l'ONU.

- **AfficherOffres** : Permet d'afficher toutes les offres du marché.

Dans ce nouveau système le représentant détient un rôle central dans le bon fonctionnement de celui-ci. Nous avons donc ajouté un nouveau attribut et trois nouvelles fonctions. Il faut savoir que maintenant un représentant peut connaître plusieurs bulletins et un marché.

## **Attribut du représentant**

- **droitVoter** : Cet attribut détermine si le représentant a le droit de voter. En effet, si le représentant vend son vote ou si il le met sur le marché, alors il perd son droit de vote.

## **Méthodes du représentant**

- **consulterMarcheVote** : Permet de consulter la liste des pays vendeurs de vote.

- **vendreVote** : Permet au représentant de mettre en vente le bulletin de vote de son pays via le marché.

- **acheterVote** : Permet au représentant d'acheter un ou plusieurs votes sur le marché

Ici, un exemple : il n'y a pas de style différent entre le titre de la sous-section (« attribut de marché »), et les éléments en dessous.

La lecture s'en retrouve plus difficile, car on a du mal à comprendre ce que regroupent les sous-sections.

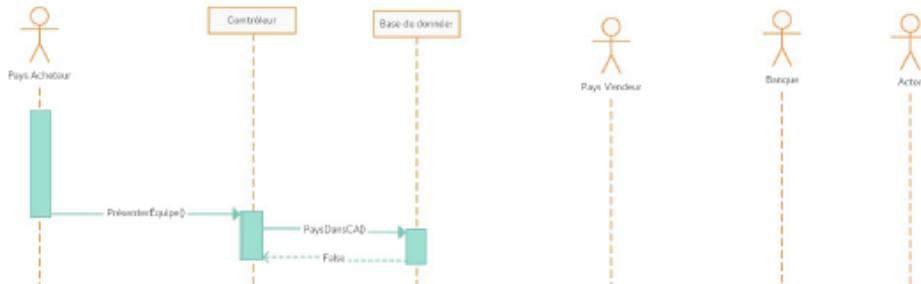




# Présentation du rapport

## Diagramme de communication et de séquence

voici les diagrammes de séquence et de communication correspondant au scénario donné dans l'énoncé : un pays propose son équipe au poste pour lequel il faut voter. Ensuite il se connecte à la liste des pays vendeur et achète un vote. Lors de la session de vote il affiche la liste des pays candidats et vote pour son équipe.



Ici, la police de caractère change de taille brusquement dans le paragraphe...



# Introduction

On écrit deux introductions :

- Une première où l'on présente sa formation, la matière et la raison de ce rapport
- Une seconde où l'on présente rapidement le sujet, et le contenu du rapport (solution apportée, méthodes utilisées, pistes explorées). C'est un peu le « teaser » de votre rapport, qui doit donner envie de lire le document.





# Engagement de non plagiat (travail seul)

Juste après la couverture, sur la troisième page du rapport (on laisse en général le dos de la couverture blanc), vous devez faire figurer l'engagement de non plagiat **(les rapports sans cet engagement seront notés à 0)**.

## Engagement de non-plagiat

Je, soussigné(e) XXX, étudiant(e) en YYY année d'école d'ingénieur à Sup Galilée, déclare être pleinement conscient(e) que la copie de tout ou partie d'un document, quel qu'il soit, publié sur tout support existant, y compris sur Internet, constitue une violation du droit d'auteur ainsi qu'une fraude caractérisée, tout comme l'utilisation d'outils d'Intelligence Artificielle pour générer une partie de ce rapport ou du code associé.

En conséquence, je déclare que ce travail ne comporte aucun plagiat, et assure avoir cité explicitement, à chaque fois que j'en ai fait usage, toutes les sources utilisées pour le rédiger.

Fait à .... , le ....

Signature :



# Engagement de non plagiat (travail en groupe)

Juste après la couverture, sur la troisième page du rapport (on laisse en général le dos de la couverture blanc), vous devez faire figurer l'engagement de non plagiat **(les rapports sans cet engagement seront notés à 0)**.

## Engagement de non-plagiat

Nous, soussigné(e)s XXX, étudiant(e)s en YYY année d'école d'ingénieur à Sup Galilée, déclarons être pleinement conscient(e)s que la copie de tout ou partie d'un document, quel qu'il soit, publié sur tout support existant, y compris sur Internet, constitue une violation du droit d'auteur ainsi qu'une fraude caractérisée, tout comme l'utilisation d'outils d'Intelligence Artificielle pour générer une partie de ce rapport ou du code associé.

En conséquence, nous déclarons que ce travail ne comporte aucun plagiat, et assurons avoir cité explicitement, à chaque fois que nous en avons fait usage, toutes les sources utilisées pour le rédiger.

Fait à .... , le ....

Signatures :



# Contenu du rapport

Le rapport doit expliquer votre démarche de résolution du problème posé par le projet, ou de l'exercice abordé en TP.

- Vous devez structurer votre rapport de façon à ce que sa lecture soit logique. Il ne faut pas forcément suivre la chronologie des questions posées, mais suivre l'ordre qui rendra la lecture la plus claire.
- Vous devez expliquer vos solutions en les décrivant, et en expliquant pourquoi elles fonctionnent.
- Vous devez aussi expliquer vos échecs (ceux qui vous semblent intéressants), en expliquant pourquoi cela n'a pas fonctionné : de chaque échec, vous devez montrer que vous avez tiré une leçon importante.



# Contenu du rapport : le code

Si vous avez du code à mettre dans le rapport,

- Pas d'impression écran (image) de code depuis l'éditeur de code !! C'est le texte qui doit être intégré, et non une image du texte !
- Pensez à utiliser un style différent pour l'intégration du code (police de caractère comme Courier New, fond gris, etc), et utilisez un service web pour obtenir une coloration syntaxique (exemple : <https://tohtml.com>)
- Ne mettez pas tout votre code dans le rapport, ce n'est pas nécessaire... Mettez les parties intéressantes, et découpez-les en plusieurs morceaux.
- Entre chaque morceau de code, vous devrez décrire ce que fait le code.



# Contenu du rapport : le code

```
for(i=0;i<9;i++)
{
    if(ptr->Tab[i]!=0)
    {
        if(ptr->Tab[i]!=Tab[i])
        {
            faux++;
            if(joueur==1 && ptr->Tab[i]!=1)
            {
                faux++;
            }
            if(joueur==-1 && ptr->Tab[i]!=2)
            {
                faux++;
            }
        }
    }
    if(ptr->premier==tour && faux<2 && ptr->scorenext>=0)
    {
        verif=1;
    }
    // printf("%d",l->taille);
}
for(i=0;i<9;i++)
{
    Tab[i]=ptr->Tab[i];
}
return choix;
```

Ce morceau sert à vérifier que dans un tour donné la grille choisie au hasard est bien adaptée .

\*La gestion du score associé à chaque grille :

On a créé une structure appelée config :

Ici, le code a été intégré par une impression d'écran. Les caractères sont déformés, et on ne peut pas copier/coller le code depuis le rapport.

Les explications à la suite sont insuffisantes !





# Contenu du rapport : le code

## ❖ `void vider_th(tab_hachage *th)`

Cette fonction permet de libérer la table de hachage allouée.

## ❖ `maillon * copie(maillon *m)`

Cette fonction prend en paramètre un maillon `m` (qui contient une configuration qui elle-même contient une grille ainsi qu'un tableau de couleurs) et renvoie une copie de ce maillon.

Elle crée un nouveau maillon `m2` et lui affecte la grille du maillon `m`, initialise le champ **nombreCouleur**, ensuite alloue un tableau de couleurs de taille **nombreCouleur** et l'initialise par les couleurs du tableau de `m`.

Ici, les explications sont présentes, mais le code n'est pas présent ! De plus, les explications sont trop descriptives, et n'expliquent pas la façon dont l'algorithme se déroule.





# Contenu du rapport : le code

❖ La structure *configuration* contient :

```
typedef struct configuration
{
    int grille[9];
    int *couleur ;
    int nombreCouleurs;
    int fin ;
    int ouverte;
    int rotationverifie;
}configuration ;
```

- **int grille[9]** : un tableau de neuf entiers, 0, si la case est vide, 1 si la case contient un rond, 2 si la case contient une croix.
- **int \*couleur** : Un tableau dynamique, qui contient les positions des couleurs dans la grille.
- **int nombreCouleurs** : le nombre de couleur que contient le tableau de couleur.
- **int fin** : un entier qui indique si la configuration de la grille arrive à une victoire(pour la machine ou bien pour le joueur).
- **int ouverte** : Un entier qui prend 1 si une boîte est ouverte, 0 sinon.
- **int rotationverifie** : Un entier qui prend 1 si une boîte donnée est identique à une autre boîte à une rotation 90 ou bien 180.

Le code est bien camouflé dans le rapport, il ne ressort pas assez. Mettre une police de caractères spécifiques (Courier New), de la couleurs et un fond gris (et pourquoi pas un cadre) autour du code permet de le faire ressortir.



# Contenu du rapport : expliquer le code

Lorsque vous décrivez ce que fait votre code, agissez en deux étapes :

- . Tout d'abord, expliquez à quoi sert globalement ce code, que va-t-il permettre de faire, en gros, dans votre projet (par exemple, »cette fonction va permettre de trier le tableau).
- . Vous pouvez ensuite expliquer, en détail, les parties importantes de l'algorithme (la variable `i` parcourt le tableau, et on compare `tab[i]` avec `tab[i+1]`).

Evitez de commencer tout de suite les commentaires de vos morceaux de code par des détails techniques... Il faut d'abord que le lecteur ait une compréhension globale de l'utilité de ce dont vous allez lui parler.



# Contenu du rapport : expliquer le code

Faites attention à ce que le code que vous présentez soit « auto contenu ».

```
void creer_fichier(FILE* input, char filename[256], int32_t nb_bit)
{
    //Concatene le tableau de codage en une chaine de caractère
    char* tmp = calloc(257*256,1);
    int32_t c;
    strcat(tmp,codage[0]);
    for(int i=1; i<256; i++)
    {
        strcat(tmp,"\n");
        strcat(tmp,codage[i]);
    }
    //Ecrit le codage Huffman dans le fichier compressé avec le nombre de bits à écrire ensuite
    FILE* output = fopen(strcat(filename,".huffman"),"w");
    fprintf(output, "%s\n%d\n", tmp, nb_bit);
    fclose(output);
    ...
}
```

Ici, le problème est que nous ne voyons pas ce qu'est la variable `codage`... On ne peut donc pas comprendre ce code.

Et si vous décrivez une fonction, expliquez, dans le prototype, le rôle des paramètres avant de décrire les détails de la fonction.



# Contenu du rapport : illustrer le code

N'hésitez pas à illustrer votre code en montrant les étapes d'un traitement algorithmique.

N'hésitez pas non plus à mettre, avant un code important, l'algorithme de votre traitement en pseudo code. Ainsi, le lecteur aura une vision globale de ce que vous cherchez à faire.

Illustrez par des schémas simples vos structures de données.



# Contenu du rapport : les diagrammes

Si vous insérez des diagrammes dans votre rapport

- N'oubliez pas de mettre une légende à votre diagramme.
- Lors de l'export de votre rapport en PDF, assurez-vous que les diagrammes sont en résolution suffisante pour le lecteur.
- Vos diagrammes doivent être dans le rapport, pas en fichier à côté du rapport !





# Contenu du rapport : les résultats

La partie résultat est très importante, car elle permet de montrer ce que vous obtenez en conclusion de votre travail.

- . Si vous avez une interface graphique, pensez à mettre des impression d'écran (bien cadrées) de votre interface.
- . Si vous avez un programme, pensez à donner une petit notice d'utilisation de ce dernier.
- . Même si ce n'est pas explicitement demandé, vous devez présenter des résultats concrets sur votre programme : taux de réussite, temps de calcul, comparaison avec d'autres programmes connus, etc.
- . Vous devez aussi faire preuve d'esprit critique et expliquer les faiblesses et limites de votre travail.





# Contenu du rapport : l'appropriation

Vous devez montrer votre capacité à vous approprier le sujet, et en y apportant un élément original (une touche personnelle) qui n'avait pas été demandé dans le sujet.

- . Vous pouvez, dans la partie résultat, faire des mesures qui n'avaient pas été demandées, afin de montrer des éléments intéressants de votre programme (taux de réussite, temps de calcul...), et expliquer tout résultat étrange obtenu.
- . Vous pouvez mettre en place un protocole de test afin de comparer votre approche avec d'autres méthodes connues.
- . Vous pouvez, dans votre code, tenter d'utiliser une méthode différente de ce qui était demandé et comparer les résultats.
- . Vous pouvez tenter d'approfondir un élément qui a attiré votre attention dans votre projet (un comportement étrange, ou un temps de calcul inattendu).



# La conclusion

Ne négligez pas la conclusion de votre rapport, car elle clôture ce dernier et laissera une impression importante dans l'esprit du lecteur.

- Expliquez ce que vous a apporté le projet, sur un plan de compétences acquises.
- Expliquez ce que vous avez réussi à faire, et ce que vous n'avez pas réussi.
- Parlez des limites de votre programme.
- Vous pouvez, si vous pensez que ce sera constructif, critiquer certaines parties du projet qui vous semblait inutile.