

Mini Projet - IF2A

version 1.5

Table des matières

1 Description des mini projets	1
1.1 Objectifs des mini projets	1
1.2 Déroulement	2
2 Cahier des charges à valider par l'enseignant	2
3 Rendu final	2
4 Notation	3
4.1 Rapport : 7 points	3
4.2 Code : 12 points	3
4.3 Archive à rendre : 1 point	3
4.4 Rendu tardif	3
4.5 Note individuelle	3

Avertissement :

— Ce document est en version 1.5. Merci de signaler toutes erreurs rencontrées.

1 Description des mini projets

1.1 Objectifs des mini projets

L'objectif du mini projet est de développer un programme informatique répondant à un cahier des charges donné. Les contraintes à respecter sont :

- le temps imparti
- la programmation doit être orientée objet
- le programme doit contenir une IHM (avec ou sans animation)
- le programme doit contenir une partie scientifique
- chaque étudiant de l'équipe (4 étudiants maximum par équipe) doit participer au projet et être capable d'expliquer son principe et son fonctionnement.

1.2 Déroulement

Les mini projets durent 5 séances dont une en complète autonomie (la salle vous est réservée mais il n'y aura pas d'enseignant). Le temps passé en séance ne suffit pas pour mener correctement le mini projet, il est nécessaire de fournir un travail personnel (*i.e.* en dehors des séances prévues). Chaque équipe est constituée entre 3 et 4 étudiants. Le tableau suivante présente le nombre d'équipes en fonction du nombre d'étudiants dans le groupe :

Nombre d'étudiants par groupe	Nombre d'équipes de 4 membres	Nombre d'équipes de 3 membres
27	6	1
26	5	2
25	4	3
24	6	0
23	5	3
22	4	2

2 Cahier des charges à valider par l'enseignant

Un cahier des charges initial est à rendre **avant** la première séance, celui-ci doit répondre à ces questions :

- le nom des participants
- le nom du projet
- l'objectif du projet
- une présentation succincte des fonctionnalités
- éventuellement vous pouvez ajouter une ébauche d'un diagramme UML des classes
- une ébauche d'interface graphique (celle-ci peut être faite à la main)

Votre enseignant validera votre cahier des charges avant le début du mini projet ou vous demandera de le modifier (trop complexe, trop simple, manque de précision, etc.) si nécessaire.

L'évaluation de ce document sera incluse dans la note finale du projet. Ce document peut évoluer pendant votre projet (avec l'accord de l'enseignant) afin d'aboutir au document final expliqué ci dessous. Vous devez également tenir un carnet de route de votre projet et un échéancier décrivant l'évolution de votre travail au fil des semaines. Tant que la thématique du mini projet respecte les contraintes ci dessus, celle ci est libre. Vous trouverez dans ce document des propositions de cahier de charges.

3 Rendu final

Il est demandé à chaque équipe de rendre un fichier compressé contenant :

1. un document appelé **readme.txt** qui décrit le contenu de l'archive
2. le code source et les éléments nécessaires à sa compilation (images, sons, etc.)
3. un document explicatif de 6 pages maximum **au format PDF** précisant :
 - (a) le cahier des charges de votre programme (quelles fonctionnalités sont proposées à l'utilisateur ?)
 - (b) la description du problème posé (quelles sont les problèmes envisagés ?)
 - (c) le principe de l'algorithme (comment les problèmes rencontrés sont ils résolus par le programme ?)
 - (d) la bibliographie (références des documents utilisés)
 - (e) la structuration des données (diagramme UML commenté)
 - (f) suggestions d'améliorations de votre projet, bugs connus
 - (g) carnet de route et échéancier décrivant l'évolution de votre travail au fil des semaines
 - (h) le pourcentage d'implication de chaque membre de l'équipe.

Ce fichier devra être déposé sous Moodle **1 semaine après la dernière séance**.

4 Notation

Une note globale sera donnée au projet. La note individuelle sera ensuite modulée selon le pourcentage d'implication de chaque membre (cf. section 4.5).

Le barème de la note globale, donné ci dessous, est indicatif.

4.1 Rapport : 7 points

- Forme et qualité de la rédaction.
- Présence d'un diagramme UML commenté.
- Qualité de la présentation du problème et de son contexte.
- Description des algorithmes utilisés.
- Analyse des forces et faiblesses du projet et pistes d'amélioration envisagées.
- Bibliographie pertinente.

4.2 Code : 12 points

- Le code compile et fonctionne.
- Décomposition en classes et méthodes.
- Qualité de la programmation : structures de données et structures de contrôle pertinentes (par ex. utilisation du bon type de boucle...).
- Niveaux de difficulté visé et atteint : ampleur du projet, difficulté des algorithmes mis en œuvre, investissements au-delà des connaissances du tronc commun.
- Lisibilité du code (respect des conventions de codage, présence de commentaires).

4.3 Archive à rendre : 1 point

- Qualité du rendu : présence du fichier readme.txt, rangement pertinent des fichiers (par ex., code source séparé des contenus multimédia).
- Attention la taille de l'archive est limitée à 50Mo maximum.

4.4 Rendu tardif

En cas de retard dans le rendu, **1 point de pénalité sera appliqué par jour de retard** (toute tranche de 24h commencée est comptabilisée). L'heure de Moodle fait foi.

4.5 Note individuelle

À l'issu du projet, une note globale du projet sera attribuée par votre enseignant habituel. Pour votre note individuelle, celle ci sera modulée selon le pourcentage d'implication indiqué dans le rapport. Ce pourcentage d'implication doit être décidé par tous les membres et leur somme doit être égale à 100%. Ce pourcentage influera sur la moitié de la note selon cette formule :

$$Note = \frac{X}{2} (1 + P * n)$$

Où :

- X est la note globale du projet
- P votre pourcentage d'implication en %
- n le nombre d'étudiants constituant l'équipe ($n \in [3, 4]$)

Par exemple, si un projet a une note globale de 14 ($X = 14$) et que l'équipe est constitué de 4 personnes ($n = 4$) alors la note de chaque membre ($Note$) selon son taux d'implication (P) sera la suivante :

Prénom	P	$Note$
Tom	30%	15.4
Pierre	20%	12.6
Emma	10%	9.8
Cathy	40%	18.2
Moyenne	25%	14