

Consignes pour le rapport de projet S3*

Pour la rédaction de projets, il faut au moins :

1. Définir **formellement** le problème
2. Donner des détails techniques

1 Comment définir formellement le problème

1.1 Objectifs

Il faut

- donner des définitions formelles (c'est à dire mathématique) des objets que vous manipulez
- identifier un certain nombre de problèmes précis pour lesquels vous définissez
 - les entrées du problème
 - les sorties du problème

Cela vous permet ensuite d'expliquer les structures de données que vous utilisez pour stocker ces données, et vos algorithmes.

Remarque : la création de diagrammes ne vous dispense absolument pas de cette partie de définition.

1.2 Exemple

On suppose que la description informelle du projet est la suivante :

"Le projet consiste à déterminer le "niveau de grammaticalité" de phrases en collectant des réponses fournies par des joueurs. La règle du jeu est la suivante. On donne deux phrases à deux joueurs différents, et on demande à chaque joueur de déterminer laquelle leur semble la plus grammaticale. Les joueurs répondent séparément sans se consulter, et gagnent des points si leur réponse est la même. Si la même phrase a été élue "la plus grammaticale", alors son

*Ces consignes ne sont évidemment qu'un complément aux cours que vous avez reçus traitant de la rédaction de projets.

"niveau de grammaticalité" augmente. L'objectif du projet est de créer un mécanisme permettant de faire jouer un grand nombre de joueurs sur un grand nombre de phrases."

Voici maintenant un exemple d'ébauche de formalisation du problème :

"A tout instant, les données du projet sont caractérisées par :

- un ensemble de n phrases $\{p_1, \dots, p_n\}$
- un ensemble de n scores (un par phrase) $\{s_1, \dots, s_n\}$, avec $s_i \in \mathbb{N}$, l'idée étant que plus s_i est grand, plus la phrase p_i est grammaticale
- m : le nombre total de joueurs inscrits
- un ensemble de m nombres de points $\{p_1, \dots, p_m\}$, avec $p_x \in \mathbb{N}$, l'idée étant que plus p_x est grand, meilleur le joueur x est

Problème 1 : jouer une partie

- entrée : deux phrases p_i, p_j ($i \neq j$), deux joueurs x, y (x différent de y)
- sortie : récupérer la réponse r_x, r_y de chaque joueur (avec r_x et r_y dans $\{i, j\}$), et
 - si $r_x = r_y$ ($= i$ par exemple), alors augmenter p_x et p_y (selon une politique à déterminer plus tard), augmenter s_i (selon une politique à déterminer plus tard) diminuer s_j (selon une politique à déterminer plus tard)
 - si $r_x \neq r_y$, diminuer p_x et p_y (selon une politique à déterminer plus tard)

... to be continued "

Remarques :

- il manque énormément de choses, mais cela donne une idée du style à avoir. Par exemple on peut imaginer que l'on veut éviter de faire jouer deux fois le même couple de phrases à un même couple de joueurs, il faudrait donc ajouter aux données l'ensemble des couples de phrases déjà jouées par chaque couple de joueurs
- constatez que l'on adpote seulement un langage mathématique, et qu'on ne parle pas ici de structures de données "informatiques" (on explique pas comment sont encodées et stockées les données, donc on ne parle pas de tableaux/listes, des serveurs sur lesquels sont réparties les données, des structures permettant l'exécution de plusieurs parties en meme temps ..). L'explication de ces structures informatiques doit etre fournie dans les détails techniques (cf Section 2).
- vous pouvez itérer avec votre tuteur pour voir si cette formalisation est correcte

2 Détails techniques (dans le rapport technique)

2.1 Objectifs

N'hésitez pas à donner des détails techniques dans le rapport technique. Ceux-ci doivent être décrits de préférence de manière concise, mais doivent être présents néanmoins. Ces détails sont souvent donnés dans la présentation orale, mais trop souvent absents du rapport qui est pourtant le support qui perdurera après le projet et qui risque d'être repris par le tuteur et/ou d'autres étudiants.

2.2 Exemple

- dans l'exemple de la section 1, expliquer les structures de données utilisées, ainsi que les algorithmes les plus importants qui vont avec (c'est aussi ici qu'il faut expliquer où sont les données, et montrer comment vos algorithmes y accèdent)
- Si l'on utilise la bibliothèque JDOM de Java pour lire ou écrire des fichiers au format XML, il faut expliquer au moins l'intérêt (éviter de faire un compilateur capable de lire ou écrire des données structurées en arbre). On peut aussi donner une description technique de son utilisation et un exemple.
- Si l'on utilise la bibliothèque SWING de Java pour faire une interface graphique, il faut au moins décrire les différents composants utilisés dans le projet.
- Si l'on lance des commandes exécutables à partir de Java (commandes obtenues par compilation d'un l'outil interfacé qui a été codé dans un autre langage, C/C++ par exemple), il faut expliquer comment vous avez procédé pour appeler les procédures systèmes, threads, etc.