



Rapport de projet du programme fonctionnel de gestion d'une bibliothèque réalisé en Pascal

Rapport d'algorithme et programmation



Réalisé par Christe Thibaud et Plumez Arnaud dans le cadre des cours d'Algorithme et Programmation ainsi que de Communication I.

A l'attention de M. Camus et de M. Perret.

Date : 19 janvier 2017



ABSTRACT


Le présent rapport de projet, présente et détaille la réalisation d'une bibliothèque en Pascal. L'accent est mis sur une compréhension globale du projet ainsi qu'une rapide vision de celui-ci. Il présente également les problèmes rencontrés, les améliorations effectuées et celles qui méritent encore un développement.

TABLE DES MATIERES

1	Introduction.....	3
2	Le projet.....	3
2.1	Situation initiale.....	3
2.2	Résultats	3
2.3	Améliorations ajoutées	3
2.4	Améliorations possibles.....	4
2.5	Organisation et planification	4
2.6	Forces et faiblesses.....	4
3	Réalisation des unités.....	5
3.1	Unité u_livre.....	5
3.2	Unité u_adherent.....	5
3.3	Unité u_biblio.....	5
3.4	Unité p_biblio.....	6
4	Conclusion	6
5	Glossaire	7
5.1	GIT.....	7
5.2	GitHub	7
5.3	GANTT	7
6	Bibliographie.....	8
6.1	Site Web.....	8
6.2	Documents.....	8
7	Annexes.....	8
7.1	GitHub	8
7.2	GANTT	8
7.3	Protocole de tests	8
8	Déclaration d'authenticité.....	8



1 INTRODUCTION

Nous réalisons ce projet, par groupe de deux, dans le cadre du cours d'Algorithmes et Programmation. Le but du travail est de réaliser un programme de gestion pour une bibliothèque en Pascal. 



Vous trouverez la synthèse des fonctionnalités implémentées ainsi que leurs résultats dans ce rapport. Ce document montre les améliorations ajoutées par nos soins et celles qu'il serait intéressant de réaliser. Il décrit également comment nous nous sommes organisés au sein du groupe. Ce rapport n'est pas écrit dans le but de décrire, dans le détail, comment fonctionne le programme ; ceci nécessiterait des compétences techniques dans le domaine ce qui n'est pas le but recherché.

2 LE PROJET



2.1 SITUATION INITIALE

Pour nous aider à réaliser ce projet, il nous était mis à disposition divers éléments. Il nous a été donné un code initial dans lequel la plupart des fonctions étaient déjà définies. Notre travail a donc été de les compléter ainsi que de les relier entre-elles pour rendre le programme fonctionnel. Pour nous aider à réaliser cette tâche, nous possédions un document nous décrivant le fonctionnement métier désiré.

2.2 RÉSULTATS

Après de nombreux commits¹, nous sommes arrivés au bout de l'implémentation de ce programme. Le résultat est semblable à l'original qui nous a été fourni au début du projet, avec certaines différences bonifiant l'affichage des informations et des améliorations qu'on a ajoutées.

2.3 AMÉLIORATIONS AJOUTÉES

Il ne nous était pas demandé de vérifier le nombre de pages et le nombre d'exemplaires d'un livre pour qu'il soit strictement supérieur à zéro. Cependant, il nous est paru important de le réaliser pour éviter des données incohérentes. De plus, nous avons vérifié les dates saisies par l'utilisateur pour éviter qu'il saisisse des dates inexistantes. Cependant, par soucis de simplicité, le programme ne prend pas en compte les années bissextiles. Nous avons également ajouté un élément au menu principal pour permettre d'afficher l'ensemble des livres de la bibliothèque.

¹ Enregistrement et validation d'une transaction avec le logiciel de gestion de versions signifiant qu'on a terminé une partie du travail.


2.4 AMÉLIORATIONS POSSIBLES


Il reste certaines fonctionnalités qui ne sont pas indispensables au fonctionnement primaire du programme mais qui seraient judicieuses d'implémenter pour éviter des erreurs techniques. Il faudrait vérifier, lors d'un ajout d'un livre ou d'un adhérent, si une autre occurrence possède déjà le même identifiant (l'ISBN pour un livre ou le code pour un adhérent).

Il y aurait également une fonctionnalité qui serait utile à implémenter pour l'utilisateur. Il s'agit de définir la date d'emprunt d'un livre automatiquement à la date actuelle. Ceci dans le but d'éviter que l'utilisateur ait à saisir manuellement et minimiser les risques d'erreur, tout en lui laissant l'opportunité de définir une date différente si l'emprunt n'a pas été réalisé le jour de la saisie.

2.5 ORGANISATION ET PLANIFICATION

Pour réaliser ce travail qui est basé sur la participation de deux personnes, nous avons opté pour l'utilisation d'un logiciel de gestion de versions : Git (cf. Glossaire Git). Tous nos codes ont également été envoyés sur la plateforme Github (cf. Glossaire Github). Les différentes tâches ont été réparties selon les unités :


- ✔ u_biblio a été assignée à Arnaud.
-  ✔ u_adherent et u_livre ont été assignées à Thibaud.
- ✔ p_biblio a été réalisée ensemble.

Afin de ne pas être surpris par le délai de notre projet, nous avons réalisé une planification sous forme d'un diagramme de Gantt (cf Glossaire Gantt) 


2.6 FORCES ET FAIBLESSES

Ce chapitre explique brièvement les forces et faiblesses constatées durant la réalisation de ce projet. Il s'agit ici d'une liste non-exhaustive des éléments principaux à relever.

Afin de réaliser ce projet, nous avons disposé de nombreuses forces. Nos connaissances acquises durant notre précédente formation ainsi que durant le module de programmation nous ont permis d'avoir une certaine aisance lors du développement. Le langage Pascal est un langage qui est caractérisé par sa clarté et sa facilité d'utilisation. De plus, il possède de la documentation. Nous possédions également, au départ, une version exemple du programme que nous avons pu exécuter afin d'analyser le fonctionnement désiré de l'application. Nous avons la chance de réaliser ce projet par groupe de deux, ce qui nous permet de gagner en productivité.

Travailler par deux peut aussi représenter une faiblesse. Cela demande une organisation irréprochable ainsi qu'une bonne communication entre les membres du groupe 

3 RÉALISATION DES UNITÉS

Ce chapitre explique comment notre code est décomposé et explicite ce que fait chaque unité. Chacune d'elle a nécessité un développement à part entière. 

3.1 UNITÉ U_LIVRE

```

C:\WINDOWS\system32\cmd.exe - p_biblio
-- Ajouter un livre a la bibliotheque --
----- Saisie d'un livre -----
- Veuillez saisir le numero ISBN :
2710330407
- Veuillez saisir le titre :
Antigone
- Veuillez le nom de l'auteur :
Jean Anouilh
- Veuillez le nombre de pages :
128
- Veuillez le nombre d'exemplaires :
5

```

Figure 1 Ajout d'un livre dans la bibliothèque

L'unité "u_livre" permet de gérer tant la création (cf. figure 1) et la suppression d'un livre que la création et la recherche d'un emprunt. Celle-ci est composée de plusieurs fonctionnalités qui se traduisent par divers **fonctions et procédures.**

3.2 UNITÉ U_ADHERENT

"U_adherent" est une unité gérant les adhérents ; cette unité est très simple. Elle s'occupe simplement de l'ajout et de l'affichage d'un adhérent (cf figure 2). Même si cette unité ne nécessite pas beaucoup de développement, il est important de l'implémenter dans une entité à part entière pour ne pas mélanger des éléments qui n'ont aucun rapport entre eux.

```

C:\WINDOWS\system32\cmd.exe - p_biblio
-- Recherche et affichage d'adherent --
Veuillez entrer le code d'un adherent : A
|- Quiroule Pierre
|  |- Code : A
|  |- Adresse : Rue de l'hopital 15
|  |- Ville : 2000 Neuchatel - Suisse
Entrez "stop" pour arreter ou "go" pour continuer : stop

```

Figure 2 Affichage d'un adhérent

3.3 UNITÉ U_BIBLIO

```

C:\WINDOWS\system32\cmd.exe - p_biblio
Affichage de Arc Biblio :
Adresse : Espace de l'Europe NT21
Ville : 2000 Neuchatel - Suisse
Affichage des 2 adherents
|- Quiroule Pierre
|  |- Code : A
|  |- Adresse : Rue de l'hopital 15
|  |- Ville : 2000 Neuchatel - Suisse
|- Perret Hubert
|  |- Code : B
|  |- Adresse : Rue principale 54
|  |- Ville : 2000 Neuchatel - Suisse
Affichage des 1 emprunts
|- 1
|  |- Livre : Chanson douce
|  |- ISBN : 9782070196678
|  |- Auteur : Leila Slimani
|  |- Adherent : Quiroule Pierre
|  |- Code : A
Appuyez sur une touche pour continuer.

```

Figure 3 Affichage des informations de la bibliothèque

L'unité "u_biblio" permet la gestion de l'entité bibliothèque. Celle-ci se charge d'afficher les informations propres (cf. figure 3) à la bibliothèque, d'ajouter et de supprimer des livres et des adhérents ainsi que d'emprunter et de rendre des livres.

Cette unité est fortement basée sur les autres et elle est principalement utilisée dans l'unité "p_biblio". Elle possède beaucoup d'appels vers les procédures et fonctions de l'unité "u_livre" ainsi que celles de l'unité "u_adherent".

3.4 UNITÉ P_BIBLIO

L'unité "p_biblio" est centrale dans notre programme. Elle sert de routage pour les actions de l'utilisateur en appelant les bonnes procédures et fonctions selon les choix de l'utilisateur. Elle s'occupe également de la gestion du menu via lequel l'utilisateur pourra utiliser l'application.

4 CONCLUSION



Pour conclure ce travail, nous allons faire le point sur l'avancement du projet. Le développement s'est déroulé correctement sans accumuler de retard et sans rencontrer de grands problèmes. Le programme est entièrement fonctionnel, nous avons même eu le temps d'y ajouter des améliorations appréciables. Cependant, il serait intéressant de développer encore quelques améliorations que nous n'avons malheureusement pas eu le temps d'effectuer.

5 GLOSSAIRE

5.1 GIT

Git est un logiciel de gestion de versions de fichiers. Il a la particularité, par rapport à ses concurrents, d'être décentralisé ; c'est-à-dire qu'il n'existe pas qu'un seul endroit où le code est déposé, chaque contributeur possède une copie du projet sur sa machine pour que chacun travaille à son rythme tout en réduisant les risques de défaillance grâce à la redondance.

5.2 GITHUB

Selon Wikipédia, « Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git » (Wikipedia, 2016). C'est-à-dire qu'il permet d'héberger sur un serveur les codes qui sont gérés à l'aide du logiciel Git et ainsi, d'assurer une sauvegarde du code hors de nos propres ordinateurs. La plateforme nous a permis également de noter toutes les tâches à réaliser ainsi que la personne qui les réalise.

5.3 GANTT

Gantt est une méthode de planification très réputée. Elle permet de visualiser facilement les différentes tâches, leur date de début et de fin, leur chevauchement entre-elles. Cette méthode aide à déterminer la date de fin du projet ainsi que le chemin critique (suite d'opérations avec un temps de battement à zéro, si celles-ci prennent du retard, la date de fin du projet sera impactée).

1IGPT

6 BIBLIOGRAPHIE

6.1 SITE WEB

Wikipedia , GitHub, In : Wikipedia [en ligne], Modifié le 7 décembre 2016,
<https://fr.wikipedia.org/wiki/GitHub>, (consulté le 11.01.2017)



Freepik, Images de la page de garde, In : FlatIcon [en ligne],
<http://www.flaticon.com/authors/freepik>, (consulté le 12.01.2017)

6.2 DOCUMENTS

CAMUS, Fabrice, 2016. Projet Algo & Prog 1 [document pdf]. Neuchâtel : HEG Arc.

7 ANNEXES

7.1 GITHUB

Adresse du projet : <https://github.com/ArnaudPI/ProjetProg1IGPT>

7.2 GANTT

Voir document gantt.mpp (Microsoft Project)

7.3 PROTOCOLE DE TESTS

Voir document protocole_de_tests.pdf (Document pdf)

8 DÉCLARATION D'AUTHENTICITÉ

Nous attestons, Plumez Arnaud, Christe Thibaud avoir fait et rédigé personnellement ce rapport algorithme et programmation et nous attestons également ne pas avoir eu recours au plagiat et avoir consciencieusement et clairement mentionné tous les emprunts faits à autrui.

Neuchâtel, le 19.01.2017

Plumez Arnaud :

Christe Thibaud :