



RAPPORT DE PROJET T.E.R

PROJET INFORMATIQUE HLIN405

PUNYDUCK



Etudiants :

- Valentin FONTAINE
- Paul BUNEL
- Esteban BARON
- Valentin PERON
- Julien LEBARON

Année : 2019-2020

Encadrante : Anne-Elisabeth BAERT

Sommaire

1	Introduction	1
2	Technologies utilisées	3
2.1	Langages	3
2.2	Outils	3
3	Développement logiciel, Conception, Modélisation, Implémentation	4
3.1	Présentation du développement	4
3.2	Modélisation	4
3.3	Fonctionnalités de l'interface	4
3.4	Format des données et procédure d'utilisation	5
3.5	Statistique	5
4	Algorithmes et Structures de Données	6
4.1	Présentation des principales structures de données	6
4.2	Présentation des principaux algorithmes	6
4.3	Complexité théorique	6
5	Gestion du Projet	7
5.1	Organisation et planification	7
5.2	Changements majeurs	8
6	Bilan et Perspectives	9
7	Bibliographie et Annexes	10

Partie 1

Introduction

Dans le cadre du TER de notre deuxième année à la faculté des sciences de Montpellier nous avons proposé un projet s'intitulant PunyDuck. C'est une plate-forme de distribution des projets des étudiants du département informatique.

Le groupe de développement est composé de cinq personnes, Valentin FONTAINE, Paul BUNEL, Valentin PERON, Julien LEBARON et Esteban BARON. Nous sommes encadré par Mme Anne-Elisabeth BAERT.

Motivation

Le TER est un module qui apporte beaucoup aux étudiants en gestion de projet ainsi qu'en programmation. Seulement une fois terminés les projets ne sont pas valorisés et tombent dans l'oubli. Notre solution est de proposer une application permettant à chaque étudiants de déposer leurs projets pour les rendre visibles et téléchargeables par tous.

Approches

Les différentes approches faces a notre problématique

Cahier des charges

Objectifs : Créer une plateforme de distribution des projets des étudiants de la Faculté des sciences de l'Université de Montpellier.

Différentes étapes

- Mise en place d'un serveur qui servira d'intermédiaire entre les utilisateurs et la base de données.
- Création d'un framework pour faciliter la réalisation de l'application.
- Conception de l'application graphique à l'aide du framework.
- Connexion entre l'application et le serveur.
- Création d'une base de données pour stocker les comptes des utilisateurs et les projets.
- Mise en service d'un site internet permettant le téléchargement de l'application.

Cahier des charges :

I - Le serveur

- Fonctionnement asynchrone
- Héberger de manière sécurisée les données des utilisateurs.
- Les utilisateurs pourront télécharger les projets hébergés.

- Doit pouvoir gérer la plupart des erreurs de réseau, comme la coupure de la connexion lors d'un téléchargement.
- II - Le framework
- Ensemble de classes et fonctions qui serviront de base à la structure d'une nouvelle application.
 - Son utilisation doit être facile avec assez de fonctionnalités déjà disponibles pour pouvoir réduire les appels de fonctions bas niveau.
 - Permettre la plasticité des interfaces
- III - L'application
- Interface graphique permettant de naviguer facilement entre différents onglets Possibilité d'accéder aux informations de son compte et de les modifier : pseudonyme, adresse mail, âge, niveau, badges.
 - Personnalisation limitée : mode clair / sombre, position de certains éléments de la fenêtre, couleur de l'arrière plan.
- IV - La base de données
- Stocke les données relatives aux comptes des utilisateurs ainsi que les projets. Les données seront cryptées.
- V - Le site
- Une page pour télécharger l'application.
 - Design responsive et dynamique

Partie 2

Technologies utilisées

2.1 Langages

- 1 - Le choix du langage pour coder l'application a été le C/C++ pour sa fiabilité, son efficacité, et ..
- 2 - De plus le C++ nous a permis de coder en orienté objet, ainsi permettant la réalisation de l'application.
- 3 - Dans le cadre des UEs HLIN202 et HLIN302 nous avons programmer en C/C++ donc nous étions rassurés car nous avions des bases.

Pour gérer la partie réseau de notre application, nous n'avions pas besoin d'utiliser un langage puissant comme le C/C++, nous pouvions donc nous rabattre sur un langage moins performant mais plus facile d'accès. De ce fait, nous nous sommes dirigés vers le langage python, qui est un langage très simple d'utilisation, pratique pour débiter dans le domaine complexe de la programmation que représente la programmation en réseau. De surcroît il est plutôt efficace pour gérer les entrées sorties, notamment pour la lecture et l'écriture dans des fichiers, qui est une notion centrale dans notre projet.

Cependant, le langage Python de base ne permet pas de faire de la programmation en réseau, nous avons donc du choisir une des nombreuses bibliothèques disponibles permettant de faire de la programmation réseau en Python. Après réflexion, nous avons opté pour le module « asyncio » pour deux raisons : premièrement, ce module offrait une interface de programmation en réseau de haut niveau donc plus simple d'utilisation, ce qui nous arrangeait particulièrement étant donné que nous sommes parfaitement novices dans ce domaine ; Deuxièmement, le gros point fort de ce module est le fait qu'il implémente une nouvelle manière de programmer : la programmation asynchrone. La programmation asynchrone pour les entrées/sorties est une forme de programmation parallèle permettant d'exécuter d'autres parties d'un programme lorsque celui-ci est en attente d'une transmission de donnée, afin de grandement diminuer le temps d'exécution du programme.

2.2 Outils

- python -> serv
- C++ -> application
- postgresql -> bdd
- UML -> conception de l'application
- discord -> communication du groupe
- git -> sauvgarder notre travail au fur et a mesure
- Visual Studio Code -> coder seul ou a plusieurs (liveShare disponible)

Partie 3

Développement logiciel, Conception, Modélisation, Implémentation

3.1 Présentation du développement

3.2 Modélisation

LQuark
<ul style="list-style-type: none">- <i>m_{parent}</i> : <i>LQuark</i> *- <i>m_{prevSibling}</i> : <i>LQuark</i> *- <i>m_{nextSibling}</i> : <i>LQuark</i> *- <i>m_{firstChild}</i> : <i>LQuark</i> *- <i>m_{lastChild}</i> : <i>LQuark</i> *- <i>m_{childrenCount}</i> : <i>LQuark</i> *
<ul style="list-style-type: none">+ <i>parent()</i> : <i>LQuark</i>* «const»+ <i>firstChild()</i> : <i>LQuark</i>* «const»+ <i>lastChild()</i> : <i>LQuark</i>* «const»+ <i>nthChild(LQindex nth)</i> : <i>LQuark</i>* «const»+ <i>prevSibling()</i> : <i>LQuark</i>* «const»+ <i>nextSibling()</i> : <i>LQuark</i>* «const»+ <i>nthSibling(LQindex nth)</i> : <i>LQuark</i>* «const*»+ <i>childrenCount()</i> : <i>LQsize</i> «const»

LQTexture
<ul style="list-style-type: none">- <i>genTexture()</i> : <i>GLuint</i>- <i>m_{id}</i> : <i>GLuint</i>- <i>m_{texWidth}</i> : <i>GLuint</i>- <i>m_{texHeight}</i> : <i>GLuint</i>- <i>m_{format}</i> : <i>GLuint</i>- <i>m_{wrapS}</i> : <i>GLuint</i>- <i>m_{wrapT}</i> : <i>GLuint</i>- <i>m_{minFilter}</i> : <i>GLuint</i>- <i>m_{magFilter}</i> : <i>GLuint</i>
<ul style="list-style-type: none">+ «create»<i>LQTexture()</i>+ <i>getId()</i> : <i>GLuint</i> «const»+ <i>getWidth()</i> : <i>GLuint</i> «const»+ <i>getHeight()</i> : <i>GLuint</i> «const»+ <i>deleteTexture(LQTexture texture)</i> : <i>void</i>«static»

LQViewable
<ul style="list-style-type: none">n : <i>uint</i>t : <i>float</i>
«create» <i>LQViewable()</i>

LQSurface
<ul style="list-style-type: none">- <i>m_{VBO}</i> : <i>GLuint</i>- <i>m_{VAO}</i> : <i>GLuint</i>- <i>m_{FBO}</i> : <i>GLuint</i>- <i>m_{shader}</i> : <i>LQShader</i> *- <i>m_x</i> : <i>LQNumber</i>- <i>m_y</i> : <i>LQNumber</i>- <i>m_{width}</i> : <i>LQNumber</i>- <i>m_{height}</i> : <i>LQNumber</i>

LQNumber
<ul style="list-style-type: none">n : <i>int</i>

3.3 Fonctionnalités de l'interface

L'interface graphique est composé de plusieurs fenêtres, chacune spécifique. D'autant plus que l'ensemble peut être personnalisable (placement de certains bloc présent sur l'interface). Pour la partie

commune à chaque fenêtre de l'interface on a la barre des onglets contenant, dans cet ordre :

- "L'accueil", comprenant les lancements rapide d'application dernièrement utilisées ainsi que les nouveautés lié à l'application ou de certains projets.
- Les "Projets", où l'ensemble des projets mis en ligne seront affichés.
- Le "Profil" est l'endroit où l'on peut modifier ses données personnelles, sa page de profil et les paramètres de l'application.
- Le "Dépôt" vas servir à faire une demande de dépôt de projet, afin qu'il soit vérifier avant mise en ligne (pour éviter les dépôt de virus ou autre).
- Icônes basique tel que épinglé, réduire et fermer. Mais contient aussi le mode nuit (fond clair qui devient foncé).

La fonctionnalité principale étant d'aller sur la page projet, voir un qui nous intéresse ou faire une demande de dépôt pour notre projet. Maintenant cas par cas voyons les différentes interactions.

Pour l'accueil il est possible de cliquer sur les articles mis en avant enfin de pouvoir les lire en détails, mais aussi les lancements rapides des derniers projet lancer en cliquant sur l'icône en question. Pour les projets, les fonctionnalités qui changes sont les touches de tri et d'affichage ainsi que la barre rechercher. Pour la collection, c'est pareil sauf que l'on peut accéder à un menu d'action a coté de chaque projet (ressemblant à 3 petit points) :

- Désinstaller
- Emplacement
- Page projet
- Désabonner / Ne plus suivre

Ainsi que les boutons d'action, lancer qui vas démarrer l'application, et installer qui vas télécharger et installer l'application.

En ce qui concerne la page de présentation du projet, on peut interagir avec la notations (données une notes ou envoyé un commentaire) mais également un lien qui va envoyé vers un forum existant en lien avec le projet. On a notamment accès au bouton suivre pour l'avoir dans notre collection et le bouton j'aime pour aimé un projet.

Les interactions avec l'interface concernant le profil sont, la modification de fond et de l'image de profil ainsi que des autres paramètres (nom, prénom, mot de passe,...), l'accès au paramètre du client (langue,emplacement,...) et le bouton de déconnexion.

3.4 Format des données et procédure d'utilisation

3.5 Statistique

Partie 4

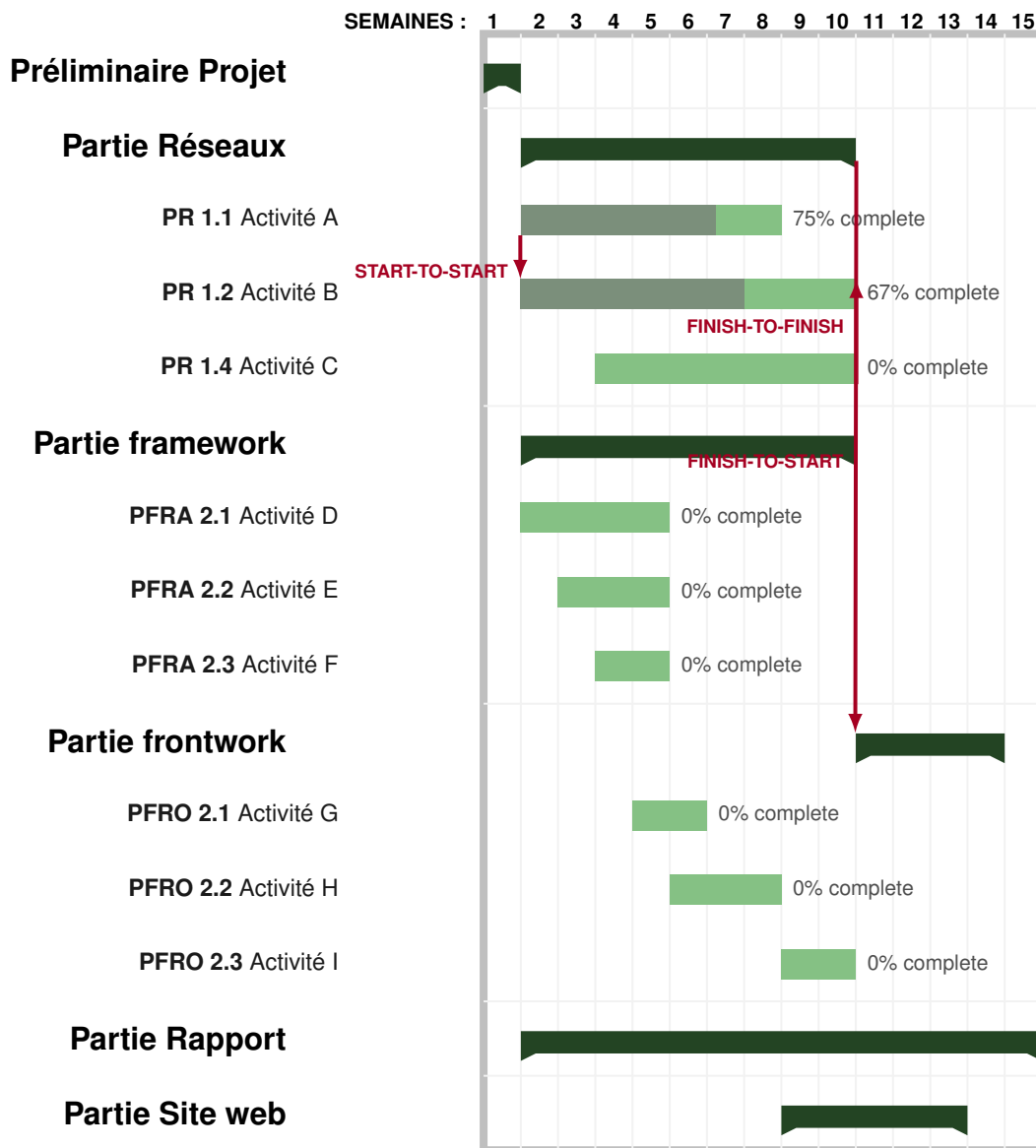
Algorithmes et Structures de Données

- 4.1 Présentation des principales structures de données
- 4.2 Présentation des principaux algorithmes
- 4.3 Complexité théorique

Partie 5

Gestion du Projet

5.1 Organisation et planification



5.2 Changements majeurs

Partie 6

Bilan et Perspectives

Le bilan de cette fin de projet est très positive, en comparant à notre cahier des charges, la quasi totalité des demandes et des objectif a été atteints. Malgré le manque de mains d'oeuvres pour la partie informatique, 3 au lieu de 5, le projet à pu voir le jour avec succès. Bilan partie réseau Bilan partie front et framework Bilan des aides (cours, site, grâce à quoi, etc...)

Conclusion

Conclusion global, point négatif et positif plus terminer par les perspectives du client.

Partie 7

Bibliographie et Annexes