Rapport de projet T.E.R

Projet Informatique HLIN405

PunyDuck

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

Encadrante : Anne-Elisabeth BAERT

Année : 2019-2020

Sommaire

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 face à notre problématique ont étaient de créer une application en deux parties. D'une part nous avons conçu l'application en elle même et d'autre part le serveur et la base de donnée. Ces deux parties communiques ensembles part le biais d'une architecture logiciel MVC (Modèle-Vue-Contrôleur). Notre appplication est téléchargeable par tous depuis le site web `https://fvostudio.com/punyduck/`.

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 débuter 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

Pour réaliser notre projet nous avons utilisé des d'outils différents et spécifique à chaque tache.

Pour l'implémentation de l'application nous sommes parti sur le langage C++ dont nous connaissions les bases et qui nous à permis de programmer en "Orienté Objet".

La partie serveur à été faite avec le langage Python pour sa facilité d'utilisation. Le domaine du réseau en informatique nous été inconnue sur le point de vue pratique, donc il nous fallait un langage

simple et puissant dans ce domaine.

La base de donnée nous avons choisi le système de gestion "PostgreSQL" pour les connaissances que nous avions sur le langage SQL du module HLIN304.

Pour la partie modélisation de l'application nous avons opté pour le Langage de Modélisation Unifié (UML) vu en cours, dans le module HLIN406.

Toutes les communications du groupe ce sont faites sur le logiciel "Discord", un logiciel facilitant grandement les communications en groupes avec par exemple le partage d'écran, les groupes vocals et textuels, le fait de pouvoir envoyer du code en un langage particulier directement dans le canal de discussion textuel etc..

Pour ce qui est des sauvegardes du projet, nous avons utilisé GIT (un logiciel de gestion de versions décentralisé) qui nous a permis de garder nos anciennes versions, de ne rien perdre en cours de route, et de pouvoir partager l'avancé du projet avec les autres étudiants du groupes et notre encadrante.

Enfin, pour ce qui est de l'éditeur de code utilisé, nous nous sommes tous penché sur Visual Studio Code pour sa fiabilité et sa mise en page agréable. De plus, nous avons pu grâce à ce logiciel coder à plusieurs en même temps avec sa fonctionnalité de partage en temps réel.

Partie 3

Développement logiciel, Conception, Modélisation, Implémentation

3.1 Présentation du développement

Le développement du projet s'est fait en deux parties, le côté application d'une part, et le côté serveur et base de donnée d'autre part.

Le développement de l'application s'est déroulé en plusieurs étapes.
La première étape fut d'apprendre l'OpenGL, un ensemble normalisé de fonctions de calcul d'images 2D ou 3D, qui nous permettrais de pouvoir créer notre propre aspect graphique car les autres outils tel que SFML, QT et gtkmm par exemples ne proposaient pas de solution précise à nos problèmes. L'OpenGL a été difficile à comprendre et à utiliser de son bas niveau comparé au langage qu'on a l'habitude de voir en L1 et L2.
La première partie de l'apprentissage d'OpenGL a été de pour dessiner un triangle d'une certaine couleur. Puis grâce à ce triangle de pouvoir dessiner un rectangle. Ainsi de suite nous avons pu dessiner toutes sortes de formes tel que les ronds et les cercles par exemple. La deuxième partie a été de pouvoir coller des textures sur ces surfaces précédemment créées. Cette partie n'a pas poser réellement de problème. Pour la troisième partie, nous devions pouvoir mettre une surface dans une autre surface comme ceci : Ceci était primordiale pour la réalisation de l'application graphique car cette simple tache

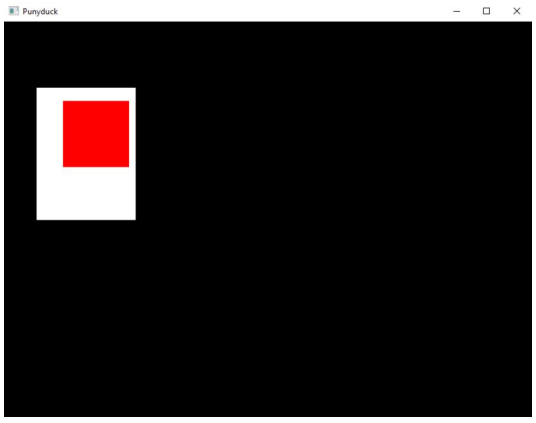


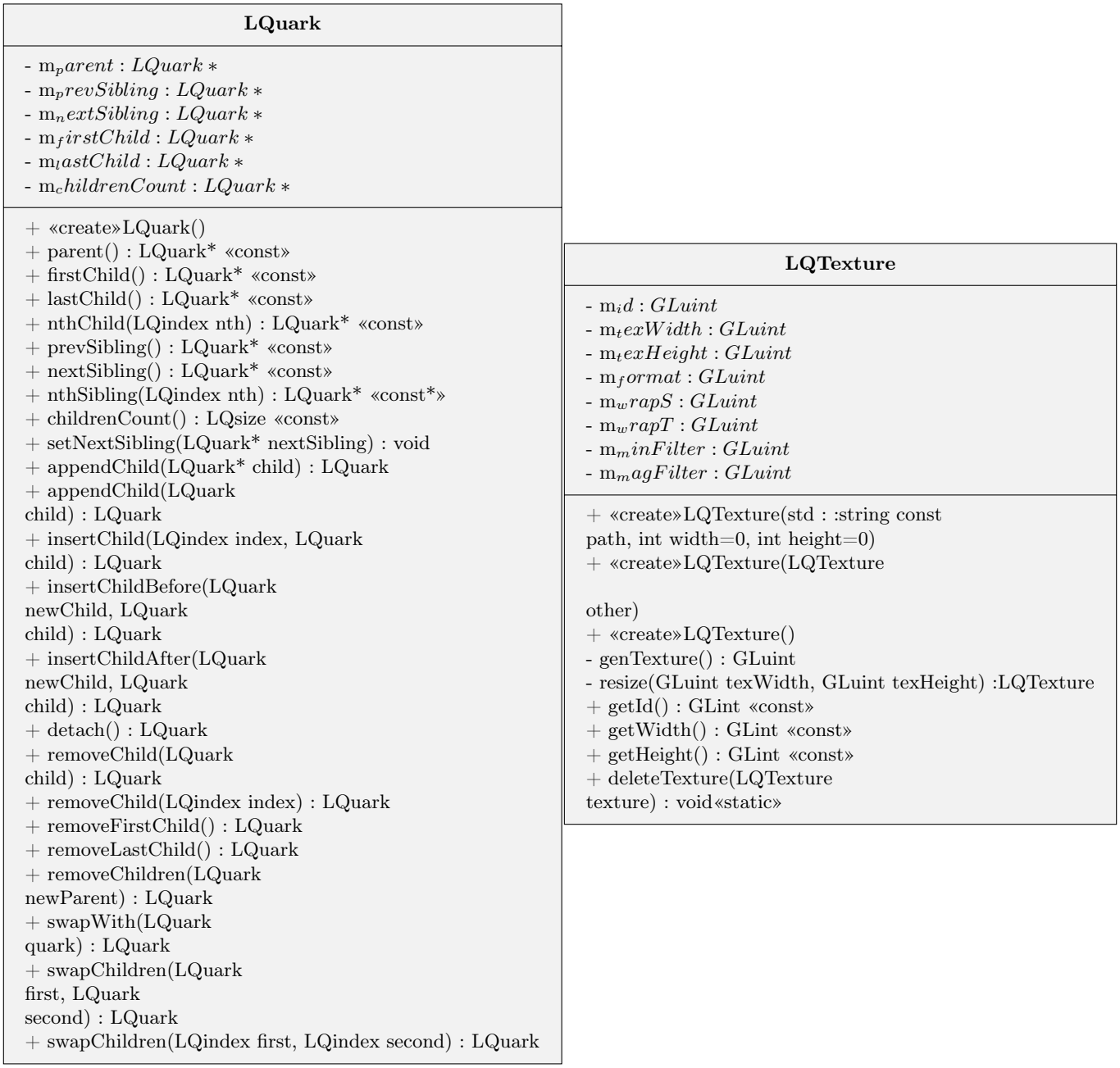
FIGURE 3.1 – Capture d'écran de notre premier essai d'une superposition de surfaces

à permis la superposition des éléments graphiques de l'application.
La dernière partie était de créer le texte, c'est-à-dire de pouvoir afficher du texte à l'écran.

La seconde étape du développement de l'application a été de modéliser en UML les différentes parties de l'application et ses mécanismes.
Pour commencer il fallait trouver un système permettant de lier des éléments entre eux afin de pouvoir les superposer, créer de nouveaux éléments à partir d'anciens, de supprimer un élément et ses sous éléments qui le compose etc.. C'est pour cela qu'on à créer les "Quark", un système ou un élément a un seul père et n fils (où n appartient à l'intervalle [0, +infini[) et a accès à ses frères (précédent et suivant). Grâce à ce mécanisme, tous les problèmes cité précédemment ont été résolus. Ce système a donc été la base de toute l'application.
Ensuite nous avons créer les différentes classes qui allaient nous servir pour l'implémentation. Comme par exemple :

- "LQTexture" qui génère les textures
- "LQSurface" pour les surfaces
- "LQViewable"
- "LQColor" pour la couleur
- "LQNumber"
- "LQText" pour le texte
- "LQButton" pour les boutons

3.2 Modélisation



3.3 Fonctionnalités de l'interface

L'interface graphique est composé de plusieurs fenêtres, chacune spécifique. D'autant plus que l'ensemble peux 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 lires en détails, mais aussi les lancements rapides des derniers projet lancer en cliquant sur l'icône en question. Pour les projets, les fonctionnalitées 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 peux acceder 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 peux 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

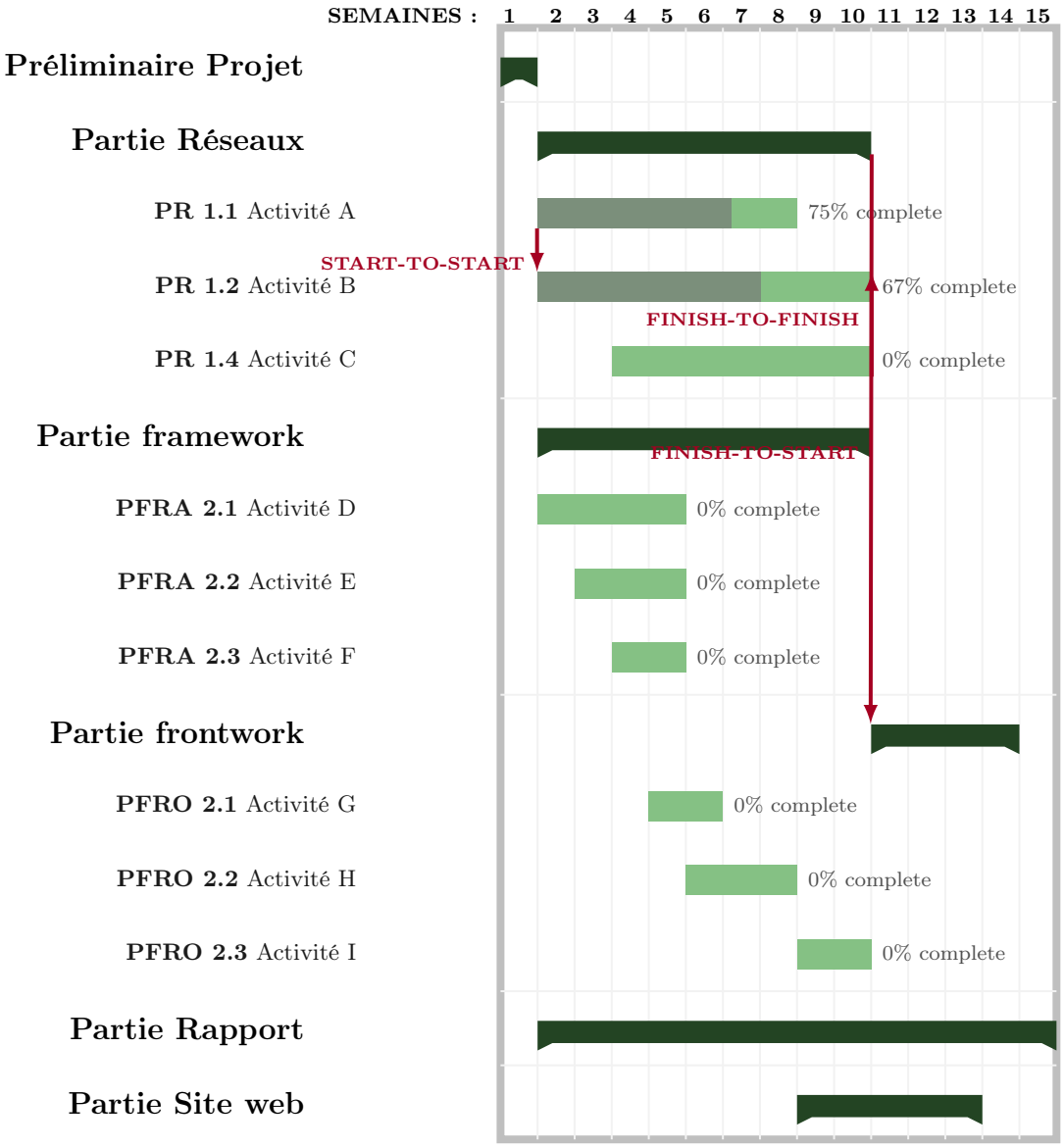
4.2 Présentation des principaux algorithmes

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