

Guillaume PEUGEOT
Pierre WAGNIER

Master 2 Informatique
2016-2017

Carnet de Bord



Encadrant : Fabrice BOUQUET

Sprint 2 : Du 18/10/2016 au 08/11/2016

1. Compte Rendu Réunion

Nous nous sommes entretenue à deux reprise avec Mr Bouquet, le mardi 18 Octobre à 16h30 et le 25 Octobre même heure, en raison de la coupure des vacances scolaires du mercredi 26 au jeudi 3 novembre. Le but de ces réunions était de présenter les outils que nous avons ainsi que les modifications que nous pensions leur apporté.

Ainsi nous avons discuté des améliorations que nous comptions à l'outil JQueryGantt. Nous avons décidé, dans un premier temps, de nous concentrer sur les améliorations essentiellement graphique et ergonomique, puis, dans un second temps, de commencer à travailler sur la liaison entre le logiciel FET et JQueryGantt.

Mr bouquet nous a expliqué l'affichage qu'il souhaitait, et qu'elle fonctionnalité était primordiale, de plus, il nous a indiqué de ne pas passer trop longtemps sur le graphique pour pouvoir travailler sur la partie liaison entre les outils.

Le prochain rendez-vous est fixé au 08/11/2016 à 16h30.

2. Objectifs

Partie Graphique :

Pour nous simplifier le travail, nous avons décidé de séparer JQueryGantt, en deux partie, la partie droite, nommé « Drawer », et la partie Gauche, nommé « Grid ».

Partie « Drawer » :

- La plage disponible est trop grande, nous devons la réduire et supprimer les horaires inutiles (de 22h à 7h).
- Les deux parties, « Grid » et « Drawer », peuvent être bouger séparément, nous devons l'empêcher.
- Nous devons ajuster l'apparence et la couleur des blocs.
- Nous devons supprimer les affichages des années et semaines tels qu'ils sont actuellement pour mettre un affichage des semaines plus synthétiques.
- Nous devons ajouter le bouton « Load », qui doit servir à charger des fichiers.
- Nous devons rajouter l'affichage de la salle, de l'encadrant, du tiers temps et gérer la multiplicité des classes par salles.
- Permettre une recherche par date dans l'emploi du temps.
- Nous devons supprimer les dépendances actuelles entre les blocs.

Partie « Grid » :

- Nous devons permettre la mise à jour des blocs « père » lors de la modifications d'un « fils ».
- Nous devons permettre l'assignement d'un encadrant à une salle/cour.
- Nous devons refaire l'affichage de l'heure, ainsi que la manière dont elle est implémentée pour une gestion.
- Nous devons protéger les entrées utilisateurs pour éviter les erreurs de format et faciliter l'insertion de donnée.
- Changer les l'affichage pour la mettre aux couleurs de l'université.
- Changer la méthode de classement des blocs.
- Trouver comment est géré la sauvegarde.
- Supprimer les dépendances codes couleurs.

Nos bases de données étant conséquentes et les opérations nombreuses, nous devons systématiquement penser à l'optimisation de nos algorithmes.

3. Travail Réalisé

Dans un premier temps, nous avons travaillé sur les parties graphiques.

Ainsi, nous avons :

- Rajouter les couleurs de l'université.
- Ajuster les couleurs/transparences des blocs.
- Proposer une solution pour gérer le tiers temps.
- Réduit et précisé les horaires.
- Ajuster les plages disponible (jour, semaine, mois)
- Commencer la gestion de l'encadrant et des salles.
- Supprimer les dépendances inutiles.
- Mise à jour automatique des « pères » par rapport au « fils » (uniquement le rajout de temps)
- Améliorer la gestion de l'heure.
- Trouver une meilleure utilisation du code couleur.

Ensuite nous nous sommes attaqués au problème pouvant présenter un problème algorithmique, c'est-à-dire la gestion du parallélisme des examens en fonction des contraintes des étudiants ainsi que la décrémentation du temps des « pères » lors de la modification de ses fils. Le problème avec le premier cas est que pour connaître les contraintes des étudiants, celle-ci étant potentiellement unique à chacun, nous sommes obligés de tous les parcourir pour valider qu'aucune contrainte n'empêche le positionnement d'un examen. Le déplacement d'un examen étant une tâche principale, celle-ci est répétée très régulièrement, et peut donc, par accumulation, prendre du temps.

Pour la modification du « père », le cas où la durée du père est augmentée est facile à gérer. Il suffit d'indiquer que la nouvelle de départ (ou de fin) est la nouvelle heure de départ du fils. Mais pour le cas de la suppression de temps chez le père, nous devons parcourir tous les fils pour trouver qu'elle le fils ayant le départ le plus tôt. La première solution était de parcourir tous les fils de récupérer leur départ, de les comparer, et de choisir l'heure correct. Mais la structure étant des filières contenant des années contenant des matières contenant des tiers temps ou autre, les parcours peuvent être longs, et la modification d'heure est une tâche principale. En discutant avec Mr Muthada Pottayya, nous sommes arrivés à une solution à base de Hashmap, permettant de raccourcir nos parcours, ainsi nous sauvegardons nos données dans plusieurs Hashmap imbriquées. Le fils ayant le temps le plus petit est toujours le premier, donc pas besoin de tout parcourir.

Nous nous sommes ensuite intéressés aux liaisons entre nos différents outils. Nous avons commencé à étudier le fonctionnement algorithmique de FET, qui est très complet ainsi que de définir les formats des données qui vont transiter entre ces outils.

4. Problème rencontré

L'interface graphique a encore besoin de beaucoup d'amélioration mais celle-ci nécessite du temps que nous préférons passer sur les parties plus complexes de notre projet. Les problèmes de parcours optimisés nous posent plus de problèmes que prévu, autant dans leur implémentation que dans leur conception. FET est un logiciel très complet, permettant beaucoup de fonctionnalités dont nous n'avons pas besoin, et de ce fait, nous devons comprendre clairement son fonctionnement pour n'en retirer que le nécessaire.