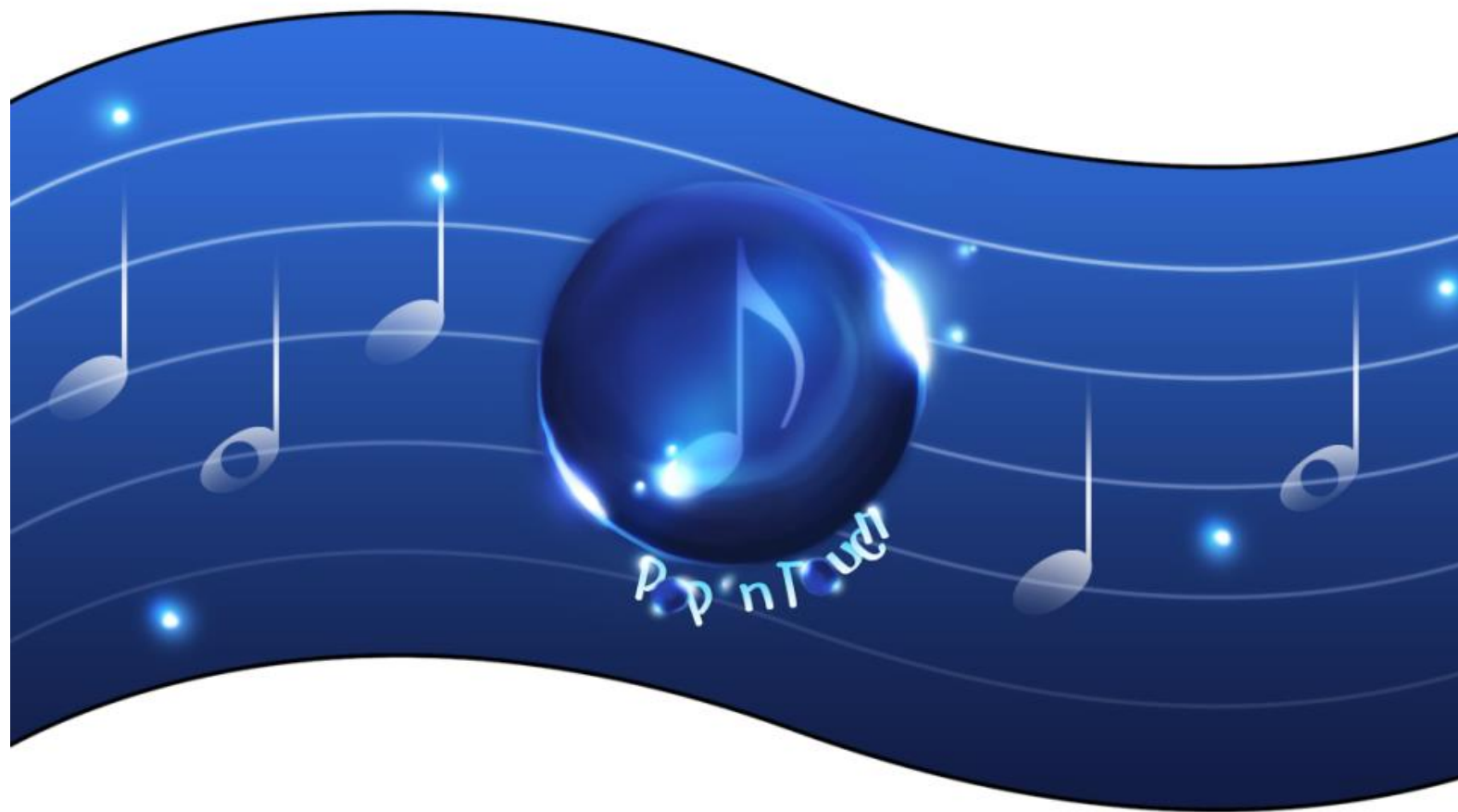


Rapport de Planification

Projet de 4^e année informatique - INSA de Rennes

Mercredi 18 Décembre 2013



Encadrants

Yann RICQUEBOURG

Cédric ALEXANDRE

Groupe

Chloé BURLOT

Nicolas BUSSENEAU

Juliette GOURLAOUEN

Gwenegon HUDIN

Damien LE GUEN

Paul-Mehdy M'RABET

SOMMAIRE

PREAMBULE	2
I. BESOIN ET CONTEXTE.....	4
A. Acteurs	4
B. Périmètre fonctionnel.....	4
C. Eléments en Entrée	4
D. Périmètre de qualification	5
E. Calendrier	5
II. ORGANISATION.....	7
A. Méthode de production en V.....	7
B. Le cycle de qualification	7
C. L'organisation.....	8
D. Le pilotage du projet.....	9
III. L'ESTIMATION	11
IV. PLANIFICATION PAR MICROSOFT PROJECT.....	12
A. Rapport de vue d'ensemble.....	12
B. Affectation des ressources par tâches	13
C. Planning.....	13
1. Jalons à respecter	13
2. Planning de conception.....	14
3. Planning de développement.....	14
D. Occupation des ressources.....	16
CONCLUSION	17
V. BIBLIOGRAPHIE	18

PREAMBULE

Pop'n'Touch 2 sera une application pour apprendre de manière accessible et ludique la lecture de partitions musicales au travers de l'interprétation de musiques. Conçue pour la table tactile PixelSense (anciennement Surface) de Microsoft, elle sera adaptée au jeu à plusieurs joueurs simultanément.

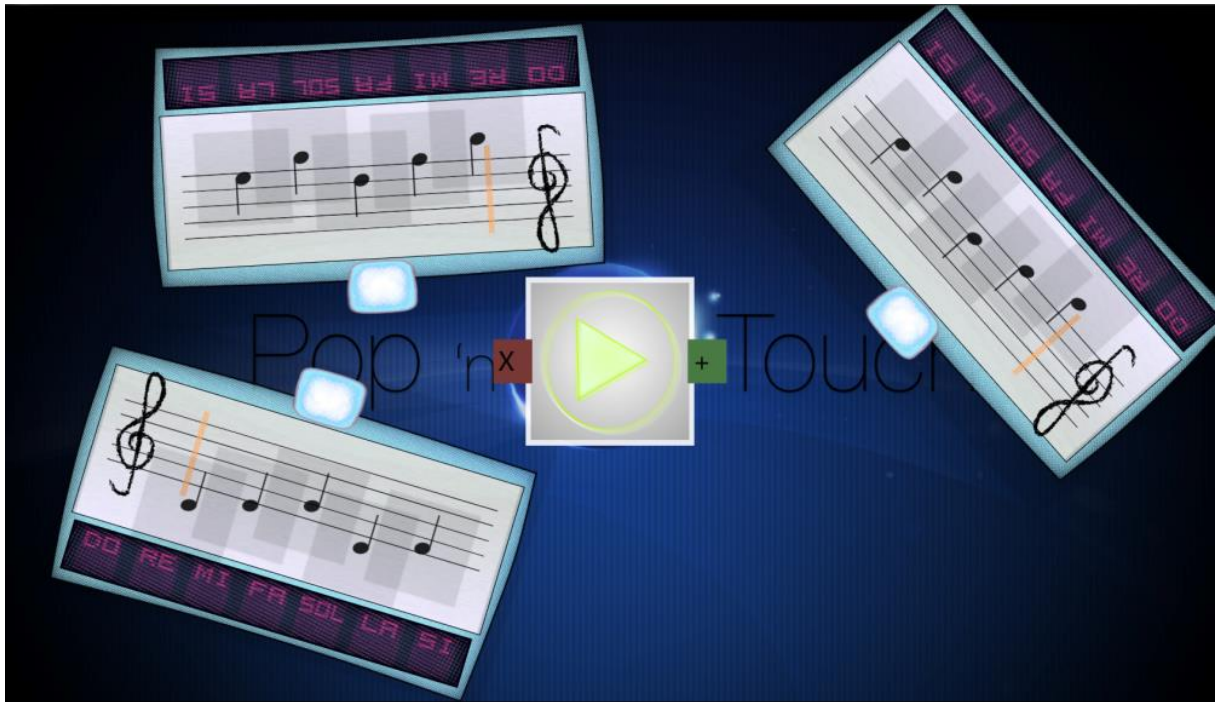


Figure 1 - Espaces de jeu de notre prototype

Pour rappeler ce qu'est Pop'n'Touch, la Figure 1 et la liste ci-dessous présentent les spécifications fonctionnelles principales de ce projet :

- La première étape (non représentée sur la capture) permet aux joueurs de choisir la chanson qu'ils souhaitent interpréter tous ensemble. Chacun d'entre eux sélectionne ensuite son instrument. En fonction du nombre de joueurs, certains instruments peuvent ne pas être interprétés du tout, ou à l'inverse être joués plusieurs fois.
- Chacun des joueurs possède son propre espace de jeu déplaçable sur la table. Pendant une partie, la partition de l'instrument qu'il aura choisi défile devant ses yeux.
- Pour pimenter le jeu, chaque joueur obtiendra de manière aléatoire des bonus, apparaissant alors dans le petit cadre en haut de son espace de jeu. Il pourra les lancer en direction d'un de ses « adversaires » pour lui compliquer la tâche.

- En fin de partie afin d'obtenir une progression de ses résultats, chaque joueur verra apparaître devant lui son score et son pourcentage de notes réussies.

L'ensemble de nos spécifications (détaillées dans le rapport précédent) nous ont permis d'aiguiller la planification de ce projet présentée dans ce rapport.

I. BESOIN ET CONTEXTE

A. ACTEURS

Ce projet, réalisé à petite échelle, ne tourne pas autour d'une myriade d'acteurs. Nous n'avons pas de client à proprement parler, ce qui nous a laissé une grande liberté d'action pour la définition du projet. Nous pouvons donc nous considérer à la fois aux commandes de la Maîtrise d'ouvrage et de la Maîtrise d'œuvre du projet Pop'n'Touch 2.

La Maîtrise d'œuvre consistera, après le départ à l'étranger de deux de nos membres, en quatre développeurs. La responsabilité de Chef de projet n'échoit définitivement à aucun d'entre nous, et sera tour à tour assumée par chacun des membres de notre équipe. Les rôles et responsabilités seront revus à chaque jalon.

Enfin, le projet dispose de deux encadrants, Yann Ricquebourg et Cédrick Alexandre, que l'on peut considérer comme parties prenantes, associés à la maîtrise d'ouvrage ou plutôt au point de vue utilisateur.

B. PERIMETRE FONCTIONNEL

Comme présenté auparavant dans les rapports de pré-étude et de spécifications, ce projet a pour but de proposer une expérience multijoueur ludique dans le cadre de l'apprentissage du solfège, en tirant pleinement parti des spécificités de la table Microsoft PixelSense.

Le projet vise donc à créer une application ludique intégrée à la table tactile, facile à prendre en main et intuitive, et permettant à travers le challenge proposé d'apprendre à lire une partition de musique. Elle ne gèrera donc aucun système d'importance critique et est pensée pour le grand public, ne nécessitant pas une très grande précision.

Le code devra être pensé pour des évolutions futures ou une reprise du projet afin d'y ajouter les spécifications secondaires ou d'autres fonctionnalités. La nécessité d'un portage de l'application sur d'autres plateformes est très peu probable, car le projet est entièrement pensé autour des particularités de son système de destination, et est réalisé pour la dernière version en date.

C. ÉLÉMENTS EN ENTRÉE

Pop'n'Touch 2 s'inscrit dans la lignée de deux précédents projets réalisés par les étudiants INFO de l'INSA de Rennes : GrooveSurface et Pop'n'Touch. Comme le nom (temporaire) de notre projet l'indique, notre principale source d'inspiration et d'éléments de réflexion est bien le projet de l'année dernière, la version 1 de Pop'n'Touch.

De celui-ci nous avons pu récupérer et étudier les rapports de pré-étude, spécifications, planification et conception. Le code source entier du projet est aussi à

notre disposition, avec ses assets (graphismes, banque sonore... cf Figure 2). Ceux-ci, avec l'application finalisée, vont nous permettre de gagner du temps dans la compréhension des technologies nécessaires au projet.



Figure 2 - Design de l'interface de Pop'n'Touch V1

D. PERIMETRE DE QUALIFICATION

Le bon fonctionnement de l'application sera vérifié principalement pour la table PixelSense sous Microsoft Windows 7 64bits, et ce par des tests en conditions réelles à chaque prototype convainquant en simulateur.

L'application doit être robuste afin de supporter les très nombreuses entrées utilisateurs erratiques qu'induit une surface tactile, et pensée pour supporter un maximum de points de contact simultanés et ce sans ralentissement, multiutilisateur oblige.

Dans les contraintes de temps qui nous sont imposées, il n'est pas envisagé d'assurer la compatibilité vers l'ancienne version de PixelSense (nommée Surface), et nous ne pouvons prévoir si l'application sera compatible avec les prochaines versions du framework Surface, qui ne sont pour l'instant pas annoncées.

E. CALENDRIER

Selon le cahier des charges de tous les projets 4INFO, ceux-ci doivent respecter des dates-clés strictes, souvent associées à un rendu de rapport ou une soutenance. Pop'n'Touch 2 ne fait bien évidemment pas exception. La phase de conception doit être terminée pour le 13 Février 2014, la construction pour le 21 mai et la livraison pour le 28 mai, date finale des projets, pour la soutenance et la démonstration.

Afin d'aider au déroulement du projet, trois périodes de cours sont bloquées et nous permettront de nous y consacrer intensivement. Les semaines du 16/12/2013, 20/05/2014 et 26/05/2014 seront donc des points culminants dans la réalisation du projet.

De plus, en période de vacances scolaires, nous allégerons la charge habituellement attribuée à chaque développeur. Les semaines de partiels, venant juste après les vacances, seront sans ressources pour Pop'n'Touch 2, car tous les développeurs seront trop concentrés à réussir leur semestre. Cela représente ainsi deux grandes périodes à l'activité restreinte pour le projet : du 22/12/2013 au 17/01/2014 et du 27/04/2014 au 16/05/2014.

La gestion du temps qui nous est attribué sera détaillée par la suite dans la présentation de notre organisation.

II. ORGANISATION

A. METHODE DE PRODUCTION EN V

Pour ce projet, nous devons produire plusieurs livrables : six rapports, une documentation en ligne, une page HTML résumant notre projet ainsi qu'un ou plusieurs exécutables, leur procédure d'installation, et enfin des jeux de tests.

Ces livrables sont dus à des dates dites « jalons », après les phases correspondantes. Par exemple, le rapport de conception sera à rendre le 13 février, après une phase de conception durant du 19 décembre au 13 février.

Ces dates « jalons » ainsi que les différentes phases sont définies de façon commune pour tous les projets : nous ne pouvons pas passer outre et devons respecter les délais imposés. C'est pourquoi nous avons préféré, après d'âpres discussions, adopter une méthode de production en V (cf. Figure 3) plutôt qu'une méthode agile avec des « sprints » qui nous semblait peu réalisable au vu de la durée probable de ces sprints (un mois). De plus, des sprints d'un mois impliqueraient de faire de la conception de façon permanente, alors que nous avons une phase de conception prévue dans la définition du projet pour cela.

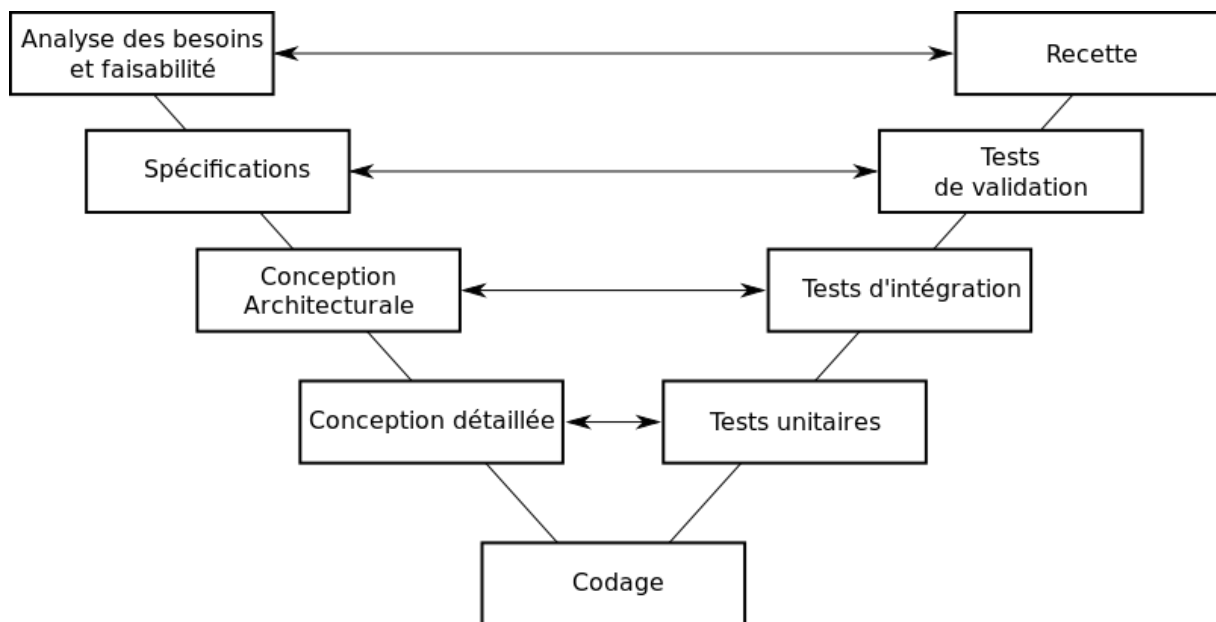


Figure 3 - Schéma d'une méthode en V

B. LE CYCLE DE QUALIFICATION

Afin de garantir que notre application reste, tout au long de son élaboration, un produit de qualité, exempt de bugs, et agréable à utiliser, nous avons choisi plusieurs approches.

Tout d'abord nous ferons usage de tests unitaires assurant qu'entre deux modifications une partie du programme ait toujours le comportement attendu. À cela s'ajouteront des tests d'intégration directement sur Microsoft PixelSense, car l'expérience des promotions précédentes sur la table tactile montre que des versions parfaitement fonctionnelles sur l'émulateur PC sont instables en situation réelle.

Nous prévoyons également, une fois qu'un prototype suffisamment élaboré et jouable sera disponible, de faire tester notre application à des personnes extérieures au projet afin d'observer leur comportement face au programme, sans leur donner une seule indication quant à son fonctionnement ni son utilisation. Cela devrait nous permettre d'identifier des vecteurs importants d'amélioration notamment sur le plan ergonomique, qui est probablement le principal pilier de notre projet.

Ce projet sera développé, au total, par quatre personnes différentes. Afin d'assurer une homogénéité en terme de syntaxe, de choix d'implémentation et de qualité de code, des créneaux de réusinage de code (*code refactoring*, cf. Figure 4) sont prévus dans notre planification. Ceux-ci reposeront principalement sur une revue, en groupe, du code produit et sur un échange de pratiques à suivre. Cette méthode permet également de rendre le code plus lisible et plus générique, facilitant ainsi son éventuelle reprise. Notons bien que lors de cette phase de refactoring, aucune fonctionnalité supplémentaire n'est implémentée.

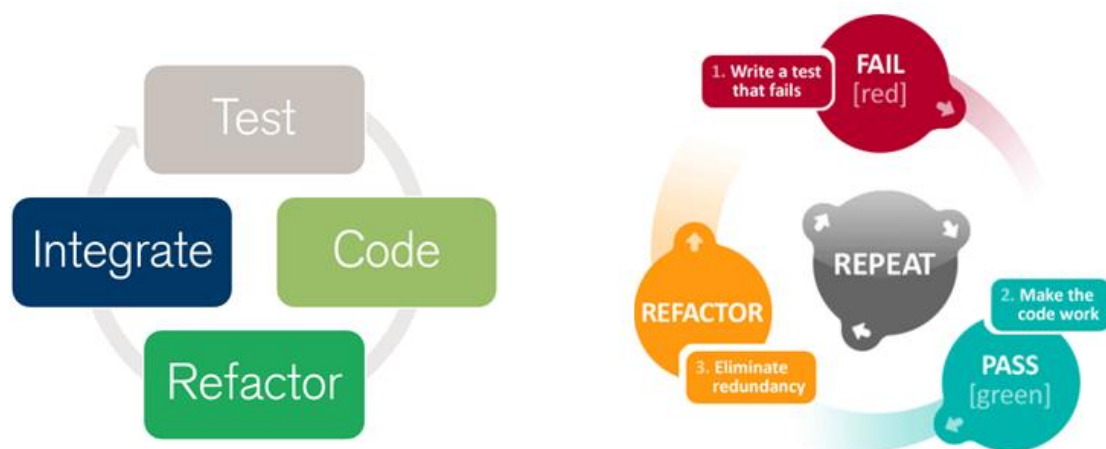


Figure 4 - Cycle de qualification

C. L'ORGANISATION

Notre groupe de travail se composera pour la partie concernée par ce rapport de 4 élèves-ingénieurs, ainsi que de Yann Ricquebourg et Cédric Alexandre en qualité d'encadrants. Étant soumis à des contraintes temporelles fortes et incompressibles de par notre statut d'étudiant, nous estimons notre capacité de travail individuelle à partir de maintenant jusqu'à la fin du projet à 7h par semaine. Ainsi, hebdomadairement le groupe pourra fournir un effort de 28h de travail.

Afin d'impliquer chacun dans une démarche de responsabilisation, nous disposons pour chaque jalon, présent dans notre calendrier, d'un référent. Celui-ci fait office de chef de projet vis-à-vis du reste du groupe pour la durée de la phase de ce jalon précis, et est donc responsable de mener à bien sa réalisation. Nos jalons étant des rapports et une soutenance finale, le chef de projet doit avant tout gérer des livrables et piloter l'équipe.

Le rôle d'architecte n'incombe à personne en particulier et est au contraire partagé entre tous les membres du groupe, puisque nous réfléchissons tous ensemble lors de la phase de conception à l'architecture du projet. C'est l'opportunité de s'approprier l'agencement du programme pour faciliter la phase de développement qui suivra juste après.

De la même manière, chacun est responsable de l'intégration et de la validation puisque nous coderons tous en parallèle, sans avoir une ressource dédiée exclusivement aux tests par exemple. Le contrôle qualité sera donc une tâche effectuée par l'ensemble de l'équipe à tout instant, afin de garantir tout au long du développement le respect du cahier des charges, de la convention de codage et de la bonne documentation du code, en plus de s'assurer de la validité du cycle de qualification.

Enfin, nous avons choisi d'utiliser les hébergeurs web GitHub et SkyDrive pour centraliser les données autour de notre projet. GitHub est hôte du code de notre application et des prototypes, tandis que SkyDrive nous permet de travailler de façon collaborative autour des rapports et de partager des documents divers. Le lien de notre dépôt GitHub est dans la bBibliographie.

D. LE PILOTAGE DU PROJET

Comme précisé plus haut, nous avons plusieurs dates jalons pour rendre nos livrables. La date la plus importante reste néanmoins celle de la soutenance pour laquelle l'application doit être finie : le 28 mai. Ayant une date butoir pour la fin de notre projet, il semble évident que ce dernier est piloté par les délais : il faut qu'il soit fini à telle date, et pas après.

Certaines tâches sont également contraintes par le fait qu'elles doivent être finies avant d'autres tâches. Ainsi, le fonctionnel de l'application doit être fait avant le visuel, les tests doivent être faits au fur et à mesure... La date de rendu des rapports contraint également les tâches qui leur sont associées : logiquement, les tâches de la planification doivent être terminées avant le rapport de planification. L'organisation des tâches se fait donc en les liants entre elles ainsi qu'aux dates « jalons » auxquelles elles sont attachées.

Le projet dispose d'une tâche hamac appelée « gestion de projet ». Les heures qui lui sont attribuées seront entièrement destinées à la conduite du projet. L'idée étant de permettre aux développeurs de faire un point de coordination sur les tâches en cours et sur les difficultés rencontrées, de synchroniser l'équipe. Cela offre aussi le

moyen de se focaliser sur la manière d'atteindre les objectifs fixés par la planification et les jalons, et de prendre du recul pour effectuer une rétrospective des tâches déjà accomplies. En cas de retard, notre équipe pourra rogner sur ces heures destinées au pilotage du projet.

III. L'ESTIMATION

Afin de définir en connaissance de cause et avec le plus de précision possible notre planification, nous avons dû estimer à la fois notre capacité de travail et la durée de chacune des tâches du projet. En théorie, l'estimation s'effectue à partir de données de références, d'éléments similaires connus et de données descriptives des tâches à estimer.

Pour calculer le temps personnel que nous pouvions allouer à ce projet nous nous sommes basés à la fois sur les volumes horaires déjà passés sur le projet jusqu'à maintenant et sur notre emploi du temps du semestre à venir. Nous avons conclu qu'en « semaine normale » nous pouvons accorder 7h de travail (soit une heure par jour ouvré hors lundi plus les trois heures de projets définies par le département, le lundi), en « semaine de projet » 25 h, en « semaine pré-partiels » et en « semaine de partiel » nous n'avons pas prévu de travailler sur le projet mais plutôt sur nos différents cours.

Pour déterminer le temps que nécessite chaque tâche, l'affaire s'avère bien plus ardue puisque c'est une activité nouvelle pour nous. Pour ce faire, nous avons utilisé les méthodes d'estimation suivantes : « Feeling » et « Analogique ».

En effet, la méthode analogique est basée sur la comparaison des tâches à chiffrer aux différentes tâches similaires que nous avons pues déjà réaliser (dans le cadre de ce projet ou non). La comparaison se fait sur la base d'une série de paramètres descripteurs.

Quatre d'entre nous ont déjà effectué au moins un stage technique dans le domaine de l'informatique. Bien qu'ayant travaillé sur des problématiques très différentes de l'actuel, nous avons pu faire appel à notre « jugement d'expert », à notre feeling pour ajouter une plus-value à notre estimation obtenue via la méthode analogique. Certains parleront même de « méthode Delphes » puisque ces mêmes membres ont confronté leurs idées jusqu'à l'obtention d'un consensus tout en limitant les influences mutuelles.

Les résultats de ces différentes estimations ont servi de base à la construction de notre planification sur Microsoft Project, disponible dans la partie suivante.

IV. PLANIFICATION PAR MICROSOFT PROJECT

A. RAPPORT DE VUE D'ENSEMBLE

Une vue macroscopique de la chronologie à venir, permet d'appréhender les grandes étapes de ce deuxième semestre de projet :

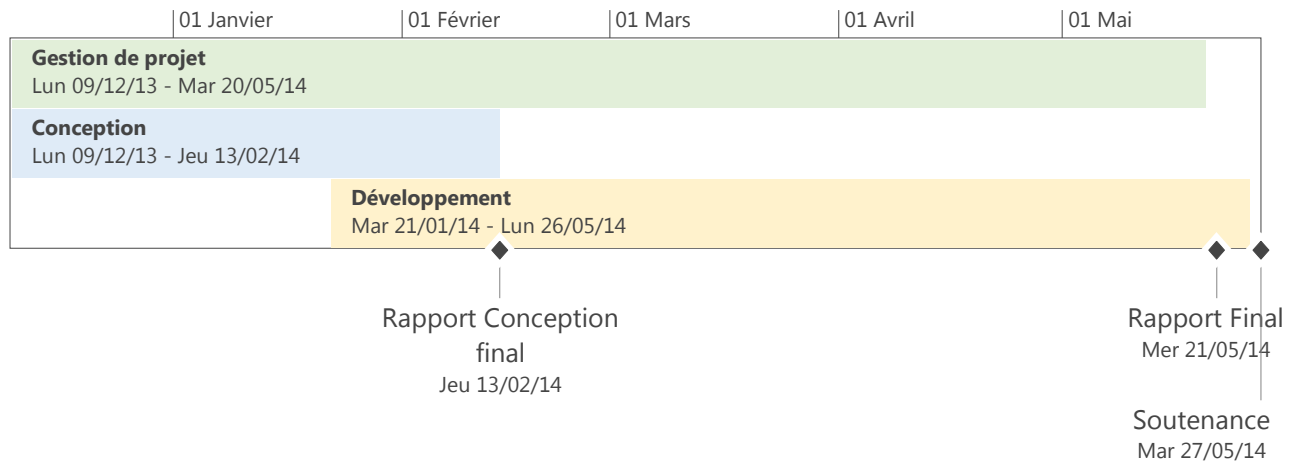


Figure 5 - Chronologie des grandes étapes

Travail

443 hr

La courbe de travail cumulé restant nous montre que la charge de travail sera globalement répartie linéairement tout au long du semestre.

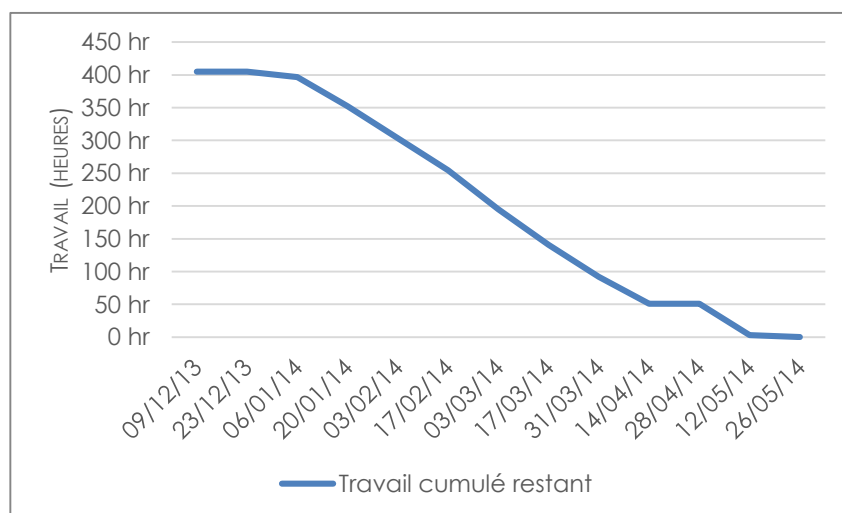


Figure 6 - Travail cumulé restant

B. AFFECTATION DES RESSOURCES PAR TACHES

Trié par ordre de date de début de groupes de tâches, le graphique ci-dessous donne le nombre de jours nécessaires pour réaliser chacun de ses groupes.

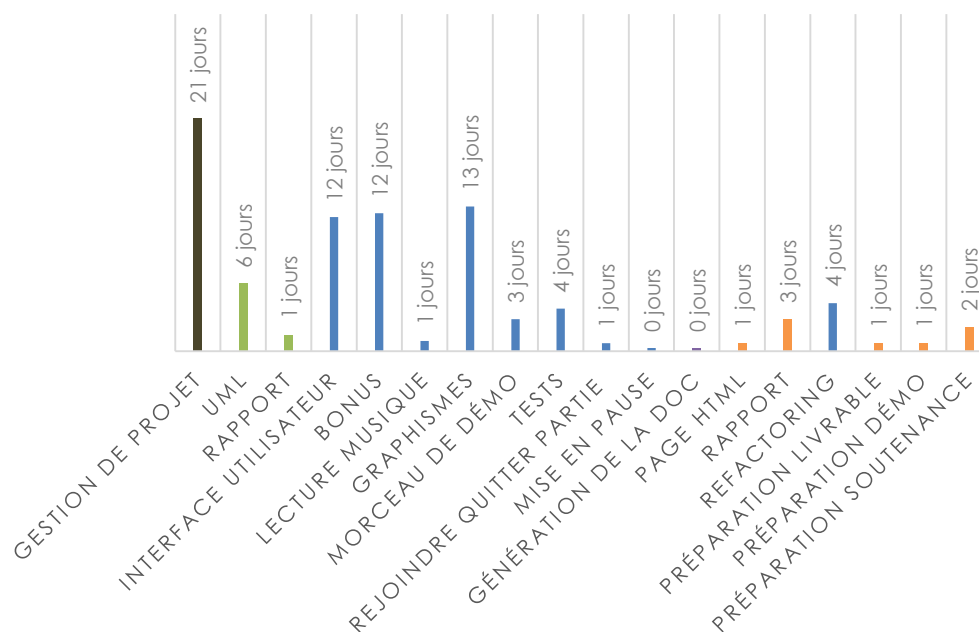


Figure 7 - Répartition du temps de travail par partie

C. PLANNING

1. Jalons à respecter

Les différentes étapes importantes du projet, telles que la remise des rapports ou la présentation du projet en soutenance, nous sont déjà imposés par avance. Voici la liste exhaustive de ces jalons :

Jalons	Date butoir
Rapport de conception V1	Ven 07/02/14
Rapport de conception Vf	Jeu 13/02/14
Page HTML V1	Mer 26/03/14
Page HTML Vf	Jeu 03/04/14
Rapport Final V1	Ven 25/04/14
Rapport Final Vf	Mer 21/05/14
Soutenance	Mar 27/05/14

2. Planning de conception

La conception, qui couvre la rédaction des diagrammes UML, s'étale de début décembre au 13 février, date du rendu du rapport de conception.

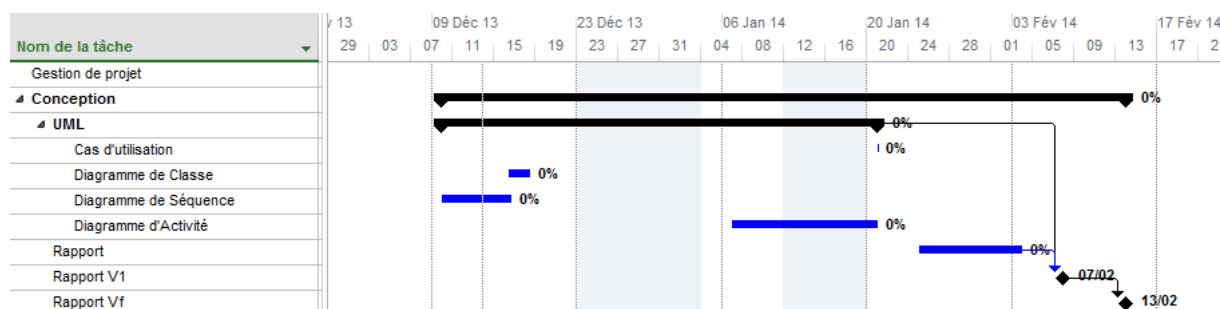


Figure 8 - Planning de conception UML

3. Planning de développement

La partie centrale du projet concerne bien évidemment le développement à proprement parlé de Pop'n'Touch. Démarrant pendant la phase de conception, il ne sera terminé qu'au moment de la rédaction de la première version du rapport final, le 25 avril. Cela nous laissera ainsi un mois pour peaufiner les fonctionnalités jusqu'à la soutenance. Le diagramme de Gantt de cette phase est présenté ci-dessous dans son intégralité, puis des zooms ont été effectués sur les parties plus spécifiques.

a) Développement général

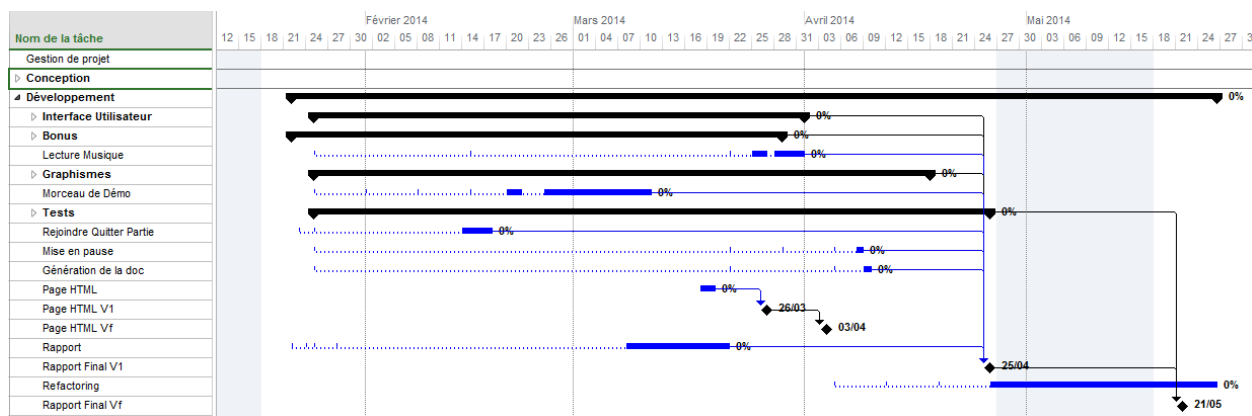


Figure 9 - Planning général

b) Interface Utilisateur et Bonus

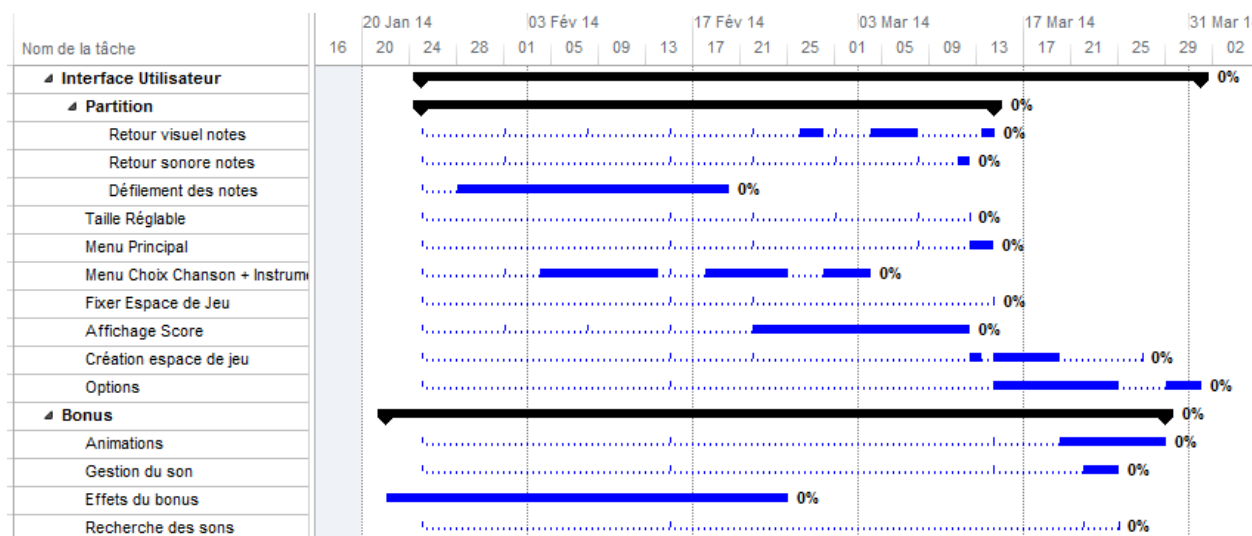


Figure 10 - Planning de développement de l'IU et les Bonus de jeu

c) Graphismes et Tests

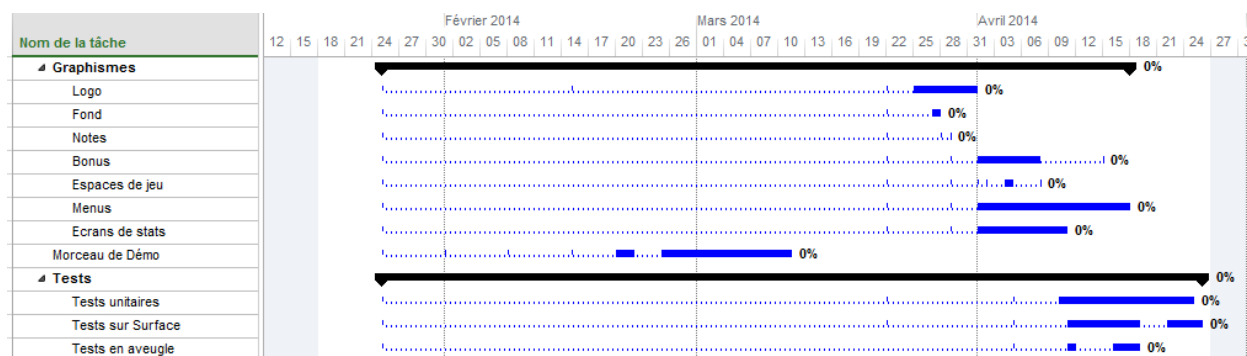


Figure 11 - Planning Infographie et Tests

4. Fin de projet

Enfin, durant le dernier mois du projet, en parallèle du réusinage sur le projet et de l'ajout éventuel de fonctionnalités facultatives, nous aurons à préparer le livrable et la soutenance de fin de projet, qui aura lieu le 27 mai.

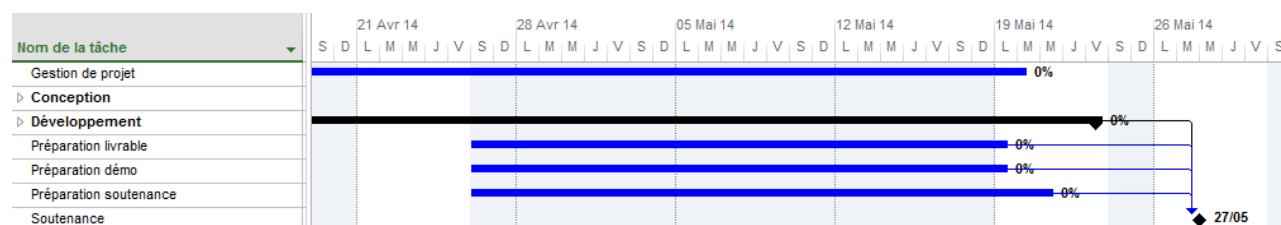


Figure 12 - Planning fin de projet

D. OCCUPATION DES RESSOURCES

Enfin nous avons effectué un audit à l'aide de Microsoft Project afin d'obtenir le pourcentage d'occupation des ressources tout au long du projet. Comme le montre le diagramme de taux d'occupation des ressources ci-dessous, à l'échelle d'une semaine nous ne serons jamais en surutilisation (ce qu'aurait signifié une barre dépassant le seuil des 100%).

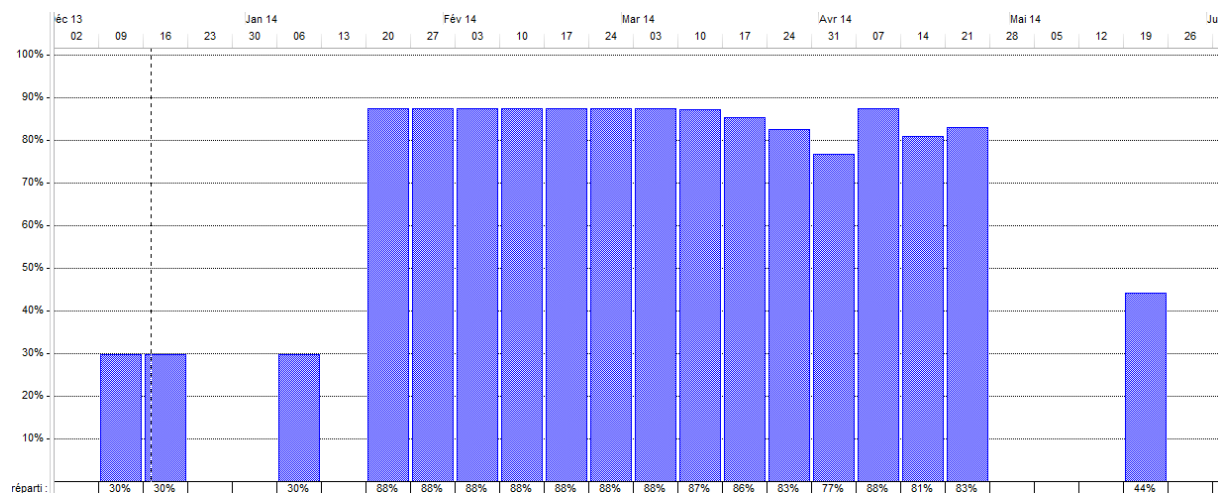


Figure 13 - Diagramme de taux d'utilisation des ressources

CONCLUSION

Nous avons donc choisi une méthode fortement inspirée du cycle de production en V communément employé afin de conduire un projet à son terme en respectant les impératifs de qualité, coût, délai et de découpage du projet en phases.

Nous avons apporté deux variantes à la méthode de base :

- Nous remplaçons la fin d'étape destinée à formaliser la validation de la phase écoulée avant de passer à la phase suivante par une tâche hamac de gestion de projet. Elle est destinée à nous apporter plus de réactivité en cas de retard ou de changement de spécification sur conseils de nos encadrants. En effet, les fins de phases étant trop éloignées les unes des autres (plusieurs semaines), il nous a semblé logique de nous affranchir de cette contrainte et d'augmenter la fréquence des réunions de gestion projet.
- Les phases de la partie montante, qui doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés, sont complétées par nos créneaux de réusinage.

Nous arrivons, d'après notre audit réalisé sur MS Project à une planification conforme aux recommandations puisqu'en moyenne nous sommes à moins de 80% d'utilisation des ressources (grain utilisé : semaine), puisque chaque jalon est respecté et que le projet sera bien rendu en temps et en heure. Notons que cela nous donne une marge de manœuvre conséquente pour adapter notre planification en cas de retard ou de force majeure.

V. BIBLIOGRAPHIE

1. Partie 4 du cours de gestion de projet : Estimation et planification par Thierry Roger
2. Code source hébergé sur GitHub :
<https://github.com/Skymirrh/Pop-n-Touch>
3. Répertoire Skydrive de documents partagés :
<https://skydrive.live.com/redir?resid=94733E909A07F469!151&authkey=!AOW0bXcOb9Gt-Jw>