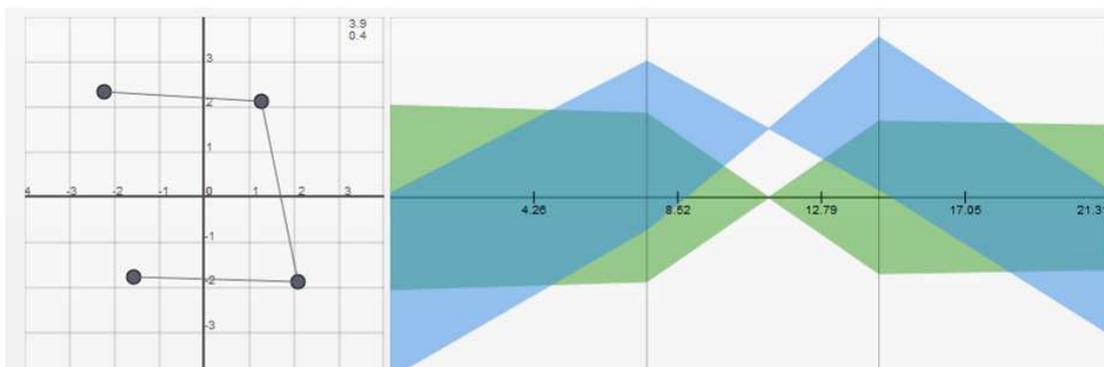


Travail de Bachelor 2012

Filière Informatique de gestion

Adaptation d'un outil pédagogique pour des tablettes



Etudiant : Loïc Haenni

Professeur : Alexandre Cotting

Mandant : Martial Geiser – HES-SO Valais

Résumé

Le but de ce travail consiste à définir la technologie la plus adaptée pour un portage de l'application sur tablettes et Web puis à réaliser la dite application.

En premier lieu, une analyse et compréhension du programme existant a dû être faite avant de définir un public cible et les objectifs du nouveau logiciel.

Une fois cette analyse faite, des choix technologiques ont dû être pris avant de pouvoir débiter le développement et l'adaptation du logiciel. Dans ces choix technologiques, il a fallu définir si nous partions sur un développement d'une application ou sur un développement d'un site web.

Après avoir choisi un développement web qui permet une utilisation tant sur ordinateurs que sur tablettes, il a fallu trouver la technologie qui permet de créer des graphiques interactifs. C'est finalement Javascript avec la librairie Raphaeljs¹ qui est choisie.

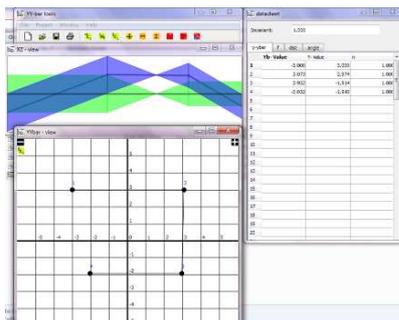


Figure 1 - Ancien programme

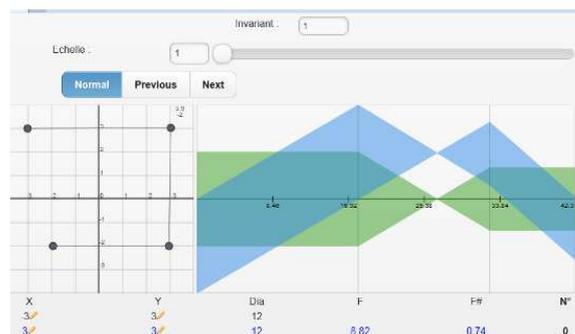
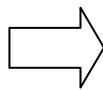


Figure 2 - Nouveau programme

Le nouveau programme permet une interactivité avec l'utilisateur grâce au diagramme de Delano. Le passage de quatre fenêtres à une seule simplifie passablement son utilisation tout en gardant toutes les informations dont l'utilisateur a besoin pour créer son système optique.

Mots-clés

Design de systèmes optiques, y-ybar, diagramme de Delano, vue de Gauss, optique de premier ordre.

¹ <http://raphaeljs.com/>

Avant-propos

L'Institut Système Industriel a développé un logiciel pour faire des simulations optiques utile tant bien dans l'éducation que pour un usage professionnel. Le logiciel a été développé de manière native pour Windows, Mac OS et Linux il y a plus de 10 ans.

L'application est toujours fonctionnelle mais peut-être améliorée suivant les nouvelles technologies et surtout pour une utilisation sur tablette. Une plus grande interactivité que la version actuelle est souhaitée.

La première difficulté a résidé dans le fait que l'optique est un domaine technique où il a fallu en apprendre les bases afin de pouvoir comprendre le fonctionnement et l'utilité du programme existant. La deuxième difficulté a été d'appliquer des formules mathématiques pour les différents calculs du site et notamment celui de « la zone rouge » qui sera expliquée plus tard dans ce document.

Ce travail a été fait en suivant les méthodes agiles avec un découpage précis des différentes tâches à faire et une planification rigoureuse sur quatre sprints différents lors du développement de la nouvelle application.

Remerciements

Par ces quelques lignes, je tiens à remercier toutes les personnes m'ayant aidé lors de la réalisation de ce travail de Bachelor et plus particulièrement :

- M. Alexandre Cotting pour le suivi global tout au long de ce travail de Bachelor et ces précieux conseils en tant que responsable de projet.
- M. Martial Geiser pour le suivi du travail « pratique », les informations, conseils et explications dans le domaine de l'optique.
- Le département optique de la HES-SO Valais pour le soutien lors du développement de l'application et la mise à disposition d'une place de travail.
- M. Eric Logean pour ses conseils sur les besoins des utilisateurs et améliorations possibles.
- M. Tilman Stuhlinger pour ses conseils sur les besoins des utilisateurs.
- M. Stéphane Geiser pour les conseils sur les graphismes et l'aspect du site web.

Déclaration sur l'honneur

Sierre, le 12 novembre 2012

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor.

Loïc Haenni

Table des matières

Adaptation d'un outil pédagogique pour des tablettes.....	i
Résumé.....	ii
Mots-clés.....	ii
Avant-propos.....	iii
Remerciements.....	iv
Déclaration sur l'honneur.....	v
Déclaration sur l'honneur.....	v
Liste des figures.....	ix
1. Introduction.....	1
2. Organisation du projet.....	2
2.1. Organisation du travail.....	2
2.2. Travail.....	2
2.2.1. Première phase : Initialisation du projet et recherche.....	2
2.2.2. Deuxième phase : Analyses.....	3
2.2.3. Troisième phase : Réalisation.....	3
2.2.4. Quatrième phase : Documentation finale.....	4
3. Analyse initiale.....	5
3.1. Logiciel actuel.....	5
3.2. Public cible.....	6
3.3. Etude de marché.....	7
3.4. Usage futur.....	8
3.5. Conclusion.....	9
4. Analyse technique.....	10
4.1. Choix de la plateforme.....	10
4.1.1. Site web.....	10

4.1.2.	Application (Androide Market – App Store)	10
4.1.3.	Conclusion.....	11
4.2.	Options utilisateurs	12
4.3.	Maquette de l'interface	13
4.4.	Détails techniques	14
4.5.	Conclusion.....	14
5.	Gestion de projet	15
5.1.	Product Backlog	15
6.	Réalisation.....	17
6.1.	Introduction.....	17
6.2.	Sprint 1	18
6.2.1.	Tâches planifiées.....	18
6.2.2.	Choix des outils de développement	18
6.2.3.	Choix de la librairie graphique.....	21
6.2.4.	Choix du langage de développement.....	22
6.2.5.	Création du site web	22
6.2.6.	Création d'un diagramme.....	23
6.2.7.	Ajout d'une lentille interactive.....	24
Bilan.....	Error! Bookmark not defined.	
6.2.8.	Tâches prioritaires pour le sprint suivant.....	25
6.3.	Sprint 2.....	26
6.3.1.	Tâches planifiées.....	26
6.3.2.	Suppression d'une lentille	26
6.3.3.	Connexion entre les lentilles	Error! Bookmark not defined.
6.3.4.	Déplacement d'une lentille	28
6.3.5.	« Zone rouge »	29
6.3.6.	Affichage des valeurs du diagramme	33

6.3.7.	Design définitif du site web	34
6.3.8.	Impression du diagramme et de ces valeurs	35
6.3.9.	Bilan	35
6.3.10.	Tâches prioritaires pour le prochain sprint	35
6.4.	Sprint 3.....	36
6.4.1.	Tâches planifiées.....	36
6.4.2.	Aperçu graphique – vue de Gauss.....	36
6.4.3.	Modification de l'invariant.....	38
6.4.4.	Point de la situation.....	38
6.4.5.	Architecture du site web.....	39
6.4.6.	Contrainte de déplacement.....	40
6.4.7.	Modification de x et y	41
6.4.8.	Modification de l'échelle du diagramme	41
6.4.9.	Echelle de la vue de Gauss	42
6.4.10.	Affichage d'une vidéo.....	42
6.4.11.	Code source protégé	43
6.4.12.	FAQ	43
6.4.13.	Déploiement du site web.....	43
6.4.14.	Bilan	44
7.	Améliorations possibles	45
8.	Retour d'expérience.....	46
9.	Conclusion.....	47
10.	Références	48

Liste des figures

Figure 1 - Ancien programme	ii
Figure 2 - Nouveau programme.....	ii
Figure 3 - Détail ancien programme	5
Figure 4 – Exemple du programme simplifié.....	6
Figure 5 - Application	7
Figure 6 - Exemple graphique HTML 5.....	11
Figure 7 - Maquette Site web	13
Figure 8 - NetBeans	19
Figure 9 - Création de projet.....	19
Figure 10 – EasyPHP	20
Figure 11 - Création d'un alias.....	20
Figure 12 - Design Mobile	22
Figure 13 - Design web	23
Figure 14 - Test largeur écran	24
Figure 15 - Création du diagramme.....	24
Figure 16 - Suppression lentille	26
Figure 17 - Message d'erreur	27
Figure 18 - Zone rouge.....	29
Figure 19 - Zone rouge calculs (1).....	30
Figure 20 - Zone rouge calculs (2).....	30
Figure 21 - Zone rouge polygone	31
Figure 22 - Code création d'une lentille	33
Figure 23 - Design standard	34
Figure 24 - Design mobile iPhone	34
Figure 25 - Vue de Gauss, ancien programme	36



Figure 26- Vue de Gauss, nouveau programme.....	37
Figure 27 - Page d'accueil.....	39
Figure 28 - Contraintes de déplacements.....	40
Figure 29 - Modification d'une valeur.....	41
Figure 30 - Modification de l'échelle exemple.....	41
Figure 31 - Echelle vue de Gauss.....	42
Figure 32 – Vidéo.....	42
Figure 33 – FAQ.....	43
Figure 34 - Release roadmap.....	46
Figure 35 - Project Velocity.....	46

1. Introduction

Le travail devait être découpé en deux phases bien distinctes. La première a consisté en une phase d'analyse tandis que la deuxième a été réservée au développement de la nouvelle application.

Durant cette phase d'analyse, la définition du cadre du travail et les objectifs ont été fait en premier. A la suite d'une entrevue avec M. Geiser et M. Cotting il a été décidé que le but du travail serait de refaire le logiciel développé nativement sous Windows.

La partie analyse, correspondant à environ 1/3 du travail, a porté dans un premier temps sur l'utilisation du programme, son public cible et une version commerciale ou non. Puis sur le choix technologique qui a aboutit sur la décision de la plate-forme du futur programme.

La deuxième phase consacrée au développement, environ 2/3 du travail, avait pour but final d'avoir d'une application interactive et fonctionnelle sur tablette reprenant toutes les fonctionnalités de l'ancien programme.

Il y avait 3 parties distinctes à développer indépendamment du choix de la plate-forme.

- Calculs
- Interface utilisateur (Diagramme de Delano)
- Graphisme (Vue de Gauss)

Ces trois parties pouvaient être implémentées en parallèle ou les unes après les autres. Par contre il a fallu définir des priorités dès le départ pour chaque tâches via le Product Backlog afin d'avoir un produit fonctionnel en fin de projet avec un maximum d'options prioritaires.

2. Organisation du projet

2.1. Organisation du travail

Ce travail de Bachelor a été partagé en deux parties au niveau des horaires de travail.

La première s'est déroulée du 17 septembre au 12 octobre. Pour un total de 140 heures soit environ 35 heures par semaines.

La seconde s'est, elle, déroulée du 15 octobre au 11 novembre. Pour un total de 220 heures soit 54 heures par semaines.

La différence importante du temps de travail entre les quatre premières semaines et les quatre suivantes s'explique par le fait d'un emploi du temps extra-scolaire très chargé en début de projet par l'organisation d'une manifestation sportive ainsi que des entraînements journaliers. Tandis que lors de la deuxième partie, tombant sur les vacances scolaires, le temps a pu être rattrapé afin d'avoir un travail de Bachelor d'environ 360 heures.

Les 4 premières semaines étant consacrées à l'initialisation du projet, de la recherche et des analyses, elles ont été placées dans un Sprint 0. Ensuite les 4 semaines restantes ont donné 4 sprints d'une semaine chacun.

2.2. Travail

2.2.1. Première phase : Initialisation du projet et recherche

Lors de cette première phase, les points suivant ont été fait :

- Prise de connaissances du projet et des directives du travail de Bachelor
- Définition des objectifs avec le mandant, M. Geiser
- Rédaction du cahier des charges
- Prise en main du logiciel actuel y-ybar
- Cours d'optique d'une journée à la HES-SO de Sion
- Création du Product Backlog suivant le cahier des charges et en discussions avec M. Geiser.
- Recherche des possibilités technologiques

Cette phase a pris 1 semaine et demie pour une durée de 50 heures

2.2.2. Deuxième phase : Analyses

Cette deuxième phase très importante avait pour but de définir deux points :

- Analyse initiale
- Analyse technique

Lors de l'analyse initiale, La définition des points suivants a été faite :

- Logiciel actuel
- Public cible
- Etude de marché
- Usage futur

Ces 4 points ont permis de donner un cadre de travail plus précis pour la suite et bien cibler les points cruciaux lors de sa réalisation grâce à l'avis de deux experts.

En ce qui concerne l'analyse technique, le choix de la plateforme a été fait. C'est ce choix qui a conditionné toute la suite du projet. L'important a été de prendre du temps et de bien discuter avec le client sur une vision à long terme afin de ne pas avoir besoin de refaire un nouveau programme dans deux ans.

Le Product Backlog a également été mis à jour suivant les décisions prises. Une recherche sur les technologies web pour la gestion graphique a également été faite.

Cette deuxième phase, qui a duré 2 semaines et demie, a pris 90 heures.

2.2.3. Troisième phase : Réalisation

C'est la phase la plus conséquente du travail puisqu'il s'agit de la réalisation du travail en lui-même. Durant cette phase, la compréhension faite lors de la phase 1 du précédent programme a été un point clé. Le gain de temps qui en a découlé ainsi que celui réalisé par un découpage précis des tâches et la définition des priorités a permis d'avoir un logiciel fonctionnel et utilisable à la fin du projet.

Cette phase a été faite de manière intensive sur 3 semaines pour une durée de 162 heures.



2.2.4. Quatrième phase : Documentation finale

La dernière semaine a été réservée pour la rédaction de la documentation finale. Cette documentation comprend ce document ainsi que la mise à jour du Product Backlog.

Cette dernière phase aura pris 54 heures.

3. Analyse initiale

3.1. Logiciel actuel

Actuellement le logiciel est composé de 4 fenêtres et se présente comme ceci :

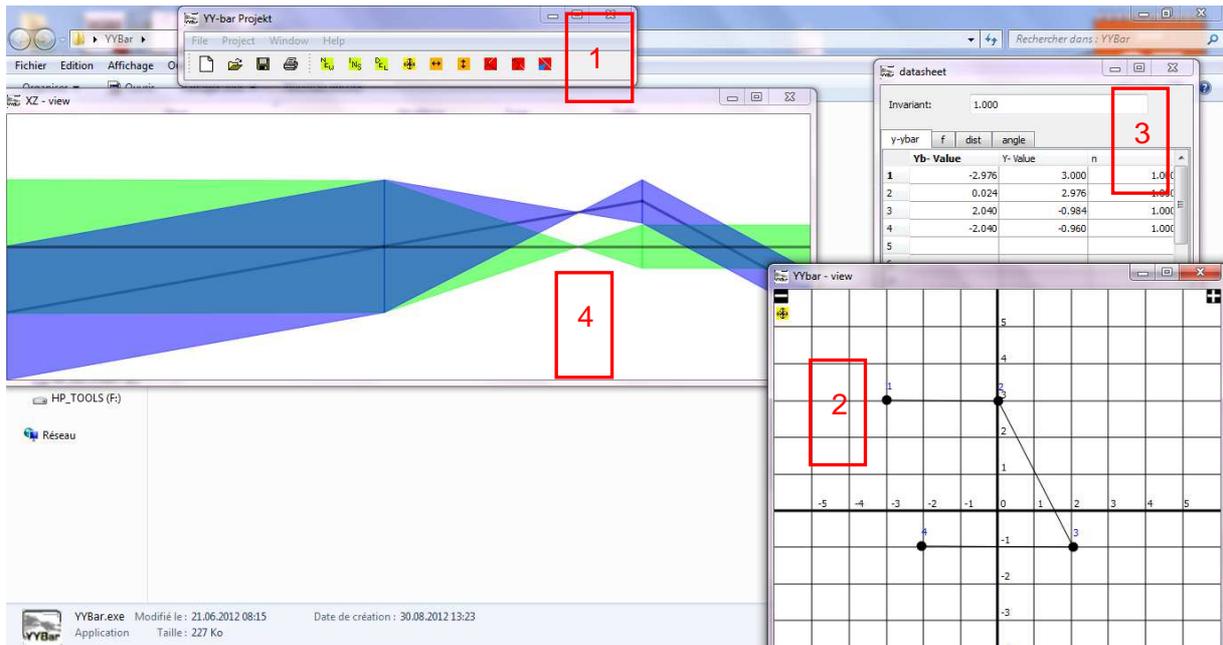


Figure 3 - Détail ancien programme

Comme nous pouvons le voir, l'écran est surchargé et ne permet pas d'afficher les 4 fenêtres sans en rogner une ou deux.

La première fenêtre est pour le menu. Dans une version future, il serait directement relié à l'interface utilisateur et non pas détaché.

La fenêtre numéro 2 est l'interface utilisateur où ce dernier va créer ses diagrammes. Tous ses paramètres sont stockés dans la fenêtre 3 où il est également possible de modifier certaines valeurs comme l'invariant.

Pour finir, nous avons une fenêtre (4) qui affiche une représentation graphique du diagramme et n'est là que pour donner un visuel du système optique à l'utilisateur.

Après discussions avec le client, un regroupement de ces fenêtres serait souhaité d'une manière suivante par exemple :

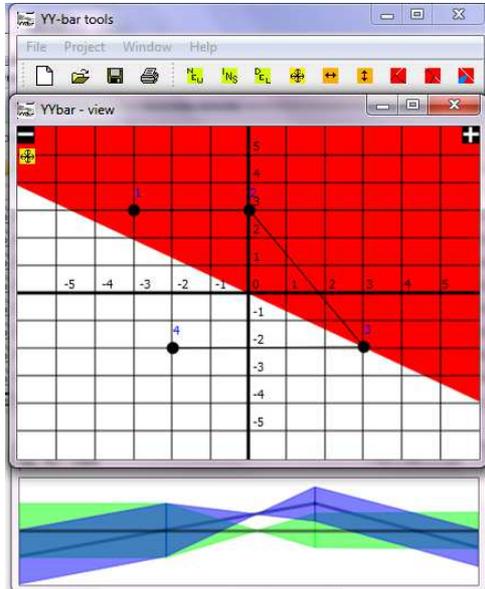


Figure 4 – Exemple du programme simplifié

Nous voyons bien dans cette exemple que l'utilisateur aurait une vision plus clair du graphique et n'aurait pas à chercher le menu sur l'écran.

La fenêtre de données (3) n'étant pas primordiale, elle peut-être masquée par défaut et ne s'afficherait que lorsque l'utilisateur le demande comme une option par exemple.

Nous voyons également sur cette image, une zone rouge qui apparaît lorsque l'utilisateur cherche à déplacer un point. Elle nous montre où il est impossible de déplacer ce point en fonction des points précédents. Il sera très important dans la nouvelle version de garder cette « zone rouge ».

3.2. Public cible

Etant donné qu'il s'agit d'un domaine très particulier et précis, ce n'est pas « Monsieur Tout-le-monde » qui s'intéressera à cette application. Donc le public cible serait restreint aux étudiants et professeurs du domaine de la physique et plus particulièrement des systèmes optiques.

Toutes entreprises dans le système industriel voulant créer rapidement un système optique devra passer par un diagramme de Delano afin d'avoir une première idée de son système et de ses valeurs. Souvent personne dans l'entreprise ne s'y connaît assez en système optique et afin de réduire des coûts d'engagement d'une personne externe, une telle application pourrait être très intéressante pour eux.

3.3. Etude de marché

Actuellement il existe déjà une application iPhone traitant le diagramme de Delano. Celle-ci comporte malgré tout un gros défaut puisqu'il n'y pas de « zone rouge » et que l'utilisateur peut créer des schémas impossible dans la réalité. Il n'y a pas non plus de possibilité d'avoir un aperçu graphique.

y-ybar Calculator

By ElyogiApps

Open iTunes to buy and down



View In iTunes

This app is designed for both iPhone and iPad

\$3.99

Category: Education
Updated: Apr 20, 2012
Version: 1.71
Size: 1.3 MB
Language: English
Seller: Ali Lopez
© 2011 ElyogiApps
Rated 4+

Figure 5 - Application iPhone y-ybar

Actuellement l'application ne figure pas dans les 300 « tops » applications payantes dans la catégorie de l'enseignement. En effet, comme il s'agit d'un domaine fermé, les profits sont moins nombreux. Dans l'enseignement, 3 catégories d'applications ressortent :

Enfants, c'est-à-dire pour aider l'enfant à l'école et/ou à son développement intellectuel

Langues, soit l'apprentissage des langues via différentes méthodes.

Culture Générale, là il s'agit « d'app » où un adulte pourrait trouver un intérêt comme un test de QI, connaissances générales, les étoiles et le ciel, théorie de du permis de conduire...

Dans les autres catégories très peu d'applications ont du succès et arrivent à ressortir du lot.

L'application y-ybar coûte 4 CHF sur l'appStore. A moins avis ce n'est pas son prix un peu plus élevé que la moyenne qui fait qu'elle ne se classe pas au « top » mais bien le fait que dans un domaine aussi précis, le nombre de « clients potentiels » est moindre.

3.4. Usage futur

Afin de pouvoir mieux comprendre les attentes d'un tel logiciel, M. Geiser m'a mis en contact avec deux personnes susceptibles d'être intéressé pas le futur logiciel et trois questions leur ont été posées :

1. Etes-vous intéressé par un tel logiciel ? Si oui, sous quelle forme (application, site web)
2. Qu'attendriez-vous le plus de ce logiciel/site web ?
3. Quel serait le point le plus important pour vous (aperçu graphique, limiter les zones où il est possible de placer une lentille, pouvoir modifier certaines variables (invariant), autres ?)

M. Eric Logean, qui avait écrit le cahier des charges de l'ancien programme, a été le premier à me répondre :

1. Oui, les deux ! Comme application pour explorer les possibilités des nouveaux écrans tactiles. Et comme site web pour la facilité d'accès et l'aspect didactique.
2. Qu'il soit facile d'utilisation: que l'on puisse modifier simplement le système optique et lire directement les valeurs de base des éléments (longueur focale, le F-number, le diamètre...)
3. Tout ! Aperçu graphique, oui, le programme doit être visuel. Limiter les zones, oui, sinon le programme va calculer quoi ? Pouvoir modifier certaines variables, oui, c'est nécessaire aussi. Tous les systèmes n'ont pas le même invariant.

M. Tilman Stuhlinger a également eu la gentillesse de prendre du temps pour répondre à ces trois questions :

1. Oui, je serais intéressé par ce logiciel comme une application.
2. Je m'attends à une interface graphique dans laquelle les lentilles peuvent être insérés et supprimés, tout en maintenant, peut-être, une distance constante entre deux objectifs existants. Il devrait également y avoir un schéma du système de lentille et une feuille de calcul indiquant les valeurs réelles de la puissance des lentilles, distance, propriétés, etc

3. La vue graphique est très important, être capable d'utiliser la souris pour déplacer des points existants dans la vue graphique. Peut-être montrer des lignes auxiliaires tels que les lignes conjuguées, des lignes d'épaisseur constante ou puissance constante, par exemple, serait également important.

Grâce à ces réponses, il a été plus facile par la suite de pouvoir définir des « User Stories » dans le Product Backlog et de pouvoir définir les priorités de développement.

Quatre points ressortent principalement de ces deux réponses :

- Qu'il y a un réel intérêt
- Que le futur programme doit être simple
- L'aperçu graphique (vue de Gauss) est essentiel
- L'utilisateur doit pouvoir accéder aux différentes données du diagramme de Delano

3.5. Conclusion

A la suite de cette analyse, J'ai décidé de ne pas partir sur un usage commercial en tout cas dans un premier temps. Le choix de la plate-forme sera le point crucial de l'analyse technique. En effet, vu le public cible restreint, un mauvais choix de plate-forme réduirait encore le nombre d'utilisateurs potentiels avec une limite technologique.

Les réponses de M. Logean et M. Stuhlinger m'ont permis de pouvoir également bien comprendre l'utilité du programme et l'orientation que devra prendre le développement.

4. Analyse technique

4.1. Choix de la plateforme

4.1.1. Site web

Avantages

- Uniquement besoin d'une connexion Internet
- Permet une utilisation sur n'importe quel support. (Ordinateurs, iPhone/iPad, Android)
- Avec un système de login, possibilité d'avoir une partie du site payante.
- L'utilisateur peut changer d'ordinateur puisque le compte est sur un serveur.
- Possibilités graphiques avec HTML5 et la balise <canvas>
- Mise à jour et maintenance facilitée.
- Interactions entre les utilisateurs possibles via un forum par exemple.
- Idéal pour une utilisation éducative car uniquement besoin d'un ordinateur. Tous les étudiants n'ont pas forcément une tablette.

Inconvénients

- Connexion Internet obligatoire. (minime grâce au 3G)
- Accès au code ? Certainement un moyen de restreindre l'accès.
- Difficile de l'intégrer à un livre.

4.1.2. Application (Androïde Market – App Store)

Avantages

- Utilisable « hors-connexion »
- Code protégé
- Possibilités graphiques avancées
- Temps de calculs plus court
- Plus « à la mode » pour des étudiants

Inconvénients

- Support limité donc tous les utilisateurs potentiels ne pourront peut-être pas l'acquérir.
- Soit entièrement gratuite, soit entièrement payante. (ou une version d'essais gratuite)
- Difficile d'apparaître dans le « top » des applications donc une visibilité restreinte.
- En cas de mise à jour, les utilisateurs doivent l'appliquer manuellement.
- Difficile de l'intégrer à un livre.
- Application iPhone déjà existante (malgré des défauts)

4.1.3. Conclusion

Chaque plateforme à ses avantages et inconvénients donc pas évident de faire un choix. Malgré tout, ce qui semble pour moi le plus important, c'est de laisser la possibilité à un maximum d'utilisateurs d'utiliser ce logiciel. C'est pourquoi le fait de n'avoir qu'une application Android ou Apple qui restreint à un support n'est pas idéal.

Grâce au site web, le diagramme sera utilisable depuis n'importe quel ordinateur ayant une connexion Internet ainsi que tous les supports tablettes équipés de 3G ou avec une connexion Wifi.

Cette solution permettant également une plus grande interactivité entre les utilisateurs et serait avantageuse dans la mesure où cela pourrait créer une communauté qui serait plus facilement en contact, (intéressant dans le cadre d'une entreprise qui cherche un spécialiste ou en cas de problèmes par exemple)

Pour les étudiants même si c'est peut-être un peu moins « design », ils pourront néanmoins y avoir accès depuis chez eux en cas de travail à domicile à faire sur le sujet. Donc l'école ne sera pas obligée de mettre à disposition des tablettes pour ce cours.

Je suis également allé voir les possibilités de HTML 5 en termes de graphismes.

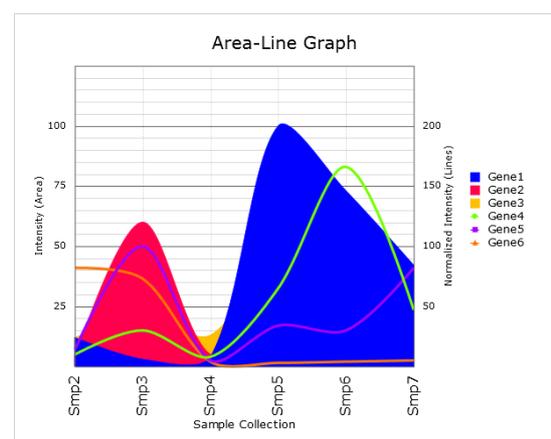


Figure 6 - Exemple graphique HTML 5

Grâce à la balise <Canvas> elles sont excellentes avec un mixage de type de courbes comme le montre l'exemple ci-dessus.

En ce qui concerne une intégration à un livre, comme dans les deux cas cela aurait été difficile de fournir l'application avec ce dernier, cela n'est pas été un critère déterminant. Pour l'intégrer il serait néanmoins possible, par exemple, de donner un QR Code qui renvoie directement sur le site et où le lecteur sera invité à s'inscrire. Un exemple d'utilisation pourra également être inclus dans le livre avec les explications.

4.2. Options utilisateurs

- L'utilisateur pourra se connecter au site avec un mot de passe.
- L'utilisateur pourra créer de nouveaux diagrammes.
- L'utilisateur pourra sauvegarder ses diagrammes et les réutiliser à sa prochaine connexion.
- L'utilisateur pourra partager ses diagrammes.
- L'utilisateur aura accès à une aide en ligne (manuels, vidéos, tutoriels)
- L'utilisateur aura accès à un forum de discussions. (Communauté)
- L'utilisateur pourra, au minimum, créer des points (lentilles), les déplacer et les supprimer.
- L'utilisateur pourra, au minimum, modifier la variable « invariant ».
- L'utilisateur pourra exporter ses diagrammes pour qu'ils soient utilisables par un autre logiciel de système optique.
- L'utilisateur pourra imprimer ses diagrammes
- L'utilisateur aura accès à toutes les données du diagramme (longueur focale, F-number, diamètre)

4.3. Maquette de l'interface

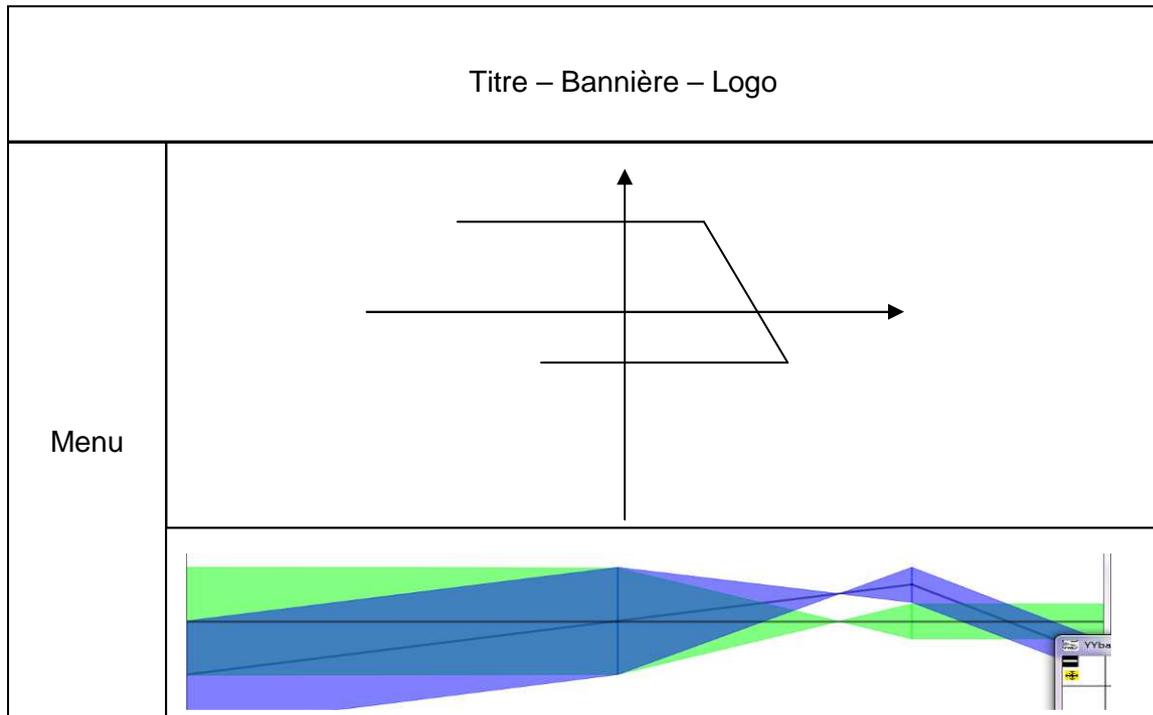


Figure 7 - Maquette Site web

- Le but est d'avoir un site rapidement pris en main et le plus simple possible pour l'utilisateur.
- Plus on ajoute de boutons/zones/textes, moins il est facile de s'y retrouver et plus long est le chargement.
- Suivant l'affichage, (tablette) le menu pourrait être horizontal par exemple ou se cacher automatiquement.
- Dans le menu, l'utilisateur pourra modifier les variables et afficher toutes les données textuelles.
- Un quadrillage serait rajouté pour le diagramme.
- Les couleurs seront plutôt claires afin de faire ressortir le graphique.
- Les boutons du diagramme seront soit à gauche soit en dessus suivant le type d'affichage.
- Une échelle de distance sur le graphique apparaîtra.
- La couleur de la lentille change suivant le rapport diamètre/distance focale.

4.4. Détails techniques

- Comme le précédent logiciel, l'utilisateur devra pouvoir, en tout cas, placer la source, les lentilles et la fin comme bon lui semble dans la mesure que cela reste réaliste. Il sera donc nécessaire d'implémenter la « zone rouge ».
- Optimiser l'affichage pour un écran tactile mais ne pas oublier l'affichage sur ordinateur. Il faudra donc créer deux styles CSS différents.
- Il faudra également protéger le code source notamment sur les calculs (Javascript ?)
- Créer une base de données pour que l'utilisateur puisse enregistrer ses paramètres et les diagrammes qu'il aura créé.
- Le forum devra être le plus simple possible tant pour le modérateur/administrateur que pour les utilisateurs.

4.5. Conclusion

Suite à la discussion avec M. Geiser, il a été décidé de partir sur une version web mais qui soit également optimisée pour un affichage mobile.

Le fait de ne pas restreindre à un type de support et l'esprit communautaire ont été les points qui ont fait pencher la balance du côté du site web.

La prochaine étape sera la création du Product Backlog avec les différentes User Stories. Les réponses de messieurs Logean, Stuhlinger et Schmidt seront intéressantes et demanderont certainement une adaptation des priorités et peut-être l'ajout de tâches suivant leurs propositions.

Le développement à proprement parlé devrait commencer lundi 15 octobre et durera jusqu'à la fin du projet. La documentation se fera tout au long du projet et les tâches devront être validées tous les jeudis par le client.

Durant le projet, l'avis des utilisateurs potentiels sera demandé afin de rester au plus proche de la réalité et ils pourront servir de beta-testeur en fin de travail également.

5. Gestion de projet

5.1. Product Backlog

15 tâches ont été définies en priorité haute. Ces tâches peuvent être découpées en trois parties :

- Site général
 - L'utilisateur doit avoir un site sobre et simple afin de pouvoir naviguer depuis son ordinateur (5)
 - L'utilisateur doit avoir un site sobre et simple afin de pouvoir naviguer depuis son smartphone ou sa tablette. (5)
 - L'utilisateur doit pouvoir naviguer entre les pages afin d'afficher toutes les pages du site. (1)
 - L'utilisateur doit pouvoir se connecter au site afin de pouvoir visualisé tout le contenu. (2)
- Diagramme interactif
 - L'utilisateur doit pouvoir créer un nouveau diagramme afin de créer un nouveau système optique. (3)
 - L'utilisateur doit pouvoir ajouter une lentille afin de simuler un système optique. (3)
 - L'utilisateur doit pouvoir déplacer une lentille afin de simuler un système optique. (3)
 - L'utilisateur doit pouvoir supprimer une lentille afin de simuler un système optique. (3)
 - L'utilisateur doit pouvoir modifier l'invariant afin de simuler un système optique. (3)
 - L'utilisateur doit pouvoir sauvegarder un diagramme afin de pouvoir le reprendre ultérieurement. (2)

- L'utilisateur doit pouvoir ouvrir un diagramme existant afin de pouvoir consulter un diagramme qu'il avait précédemment créé et l'éditer s'il le désire. (2)
- L'utilisateur doit pouvoir supprimer un diagramme afin de « nettoyer » son compte pour ne garder que les diagrammes qu'il utilise. (2)
- L'utilisateur doit pouvoir exporter un diagramme afin de pouvoir l'utiliser sous un autre logiciel d'optique. (5)
- Graphique et données
 - L'utilisateur doit avoir un aperçu graphique afin qu'il se rende mieux compte de son système optique. (10)
 - L'utilisateur doit afficher les valeurs du système optique afin de connaître les différentes valeurs du système optique et vérifier si elles sont exactes. (5)

Plusieurs tâches ont ensuite été définies avec une priorité plus basse. Il y a notamment une console d'administration afin de gérer les utilisateurs, les diagrammes et les vidéos.

Entre parenthèses, il s'agit du nombre de « Story Points » suivant une estimation de la durée de chaque tâche.

Par la suite, vous trouverez des remarques sur la valeur de ces points qui n'ont pas toujours été bien estimés et l'explication de la mauvaise estimation.

6. Réalisation

6.1. Introduction

La phase de réalisation est la partie centrale du projet. Après l'analyse technique, la décision de partir sur un développement web a été prise. Grâce au Product Backlog, le développement a été découpé en plusieurs tâches et suivant un ordre précis sur les quatre semaines.

Dans la suite du document, toutes les parties développées seront expliquées. Elles sont découpées de la même manière que le Product Backlog, c'est-à-dire par Sprint.

Dans le premier sprint, la mise en place du site, les choix technologiques et la mise en place des outils de développement ont été les points essentiels et qui ont pris le plus de temps.

Dans le deuxième sprint, un gros point a été traité. Il s'agit de la gestion du diagramme de Delano. En effet, afin d'avoir un diagramme interactif sans bugs ou erreurs, il était important de passer du temps dessus. La validation du graphisme du site a également été faite après une discussion avec une personne travaillant dans ce domaine.

Pour finir, le troisième sprint, a été géré de manière différente. En effet, au début de celui-ci, un point de situation a été fait avec le client et une redéfinition des tâches a eu lieu. L'aperçu graphique aussi appelée « vue de Gauss » avait été faite juste avant cette mise au point. Cela a permis au client de pouvoir choisir exactement ce qu'il voulait avoir pour la fin du projet. Un retour de M. Eric Logean a également été reçu ce qui a permis d'avoir un avis d'une personne externe sur ce qu'il manquait et ce qui pouvait être amélioré avant la fin du projet et dans le futur.

Plusieurs questions se sont également posées en début de réalisation telle que le langage de développement, la réutilisation du code source du précédent programme ou la librairie graphique à utiliser. Ces questions trouveront réponses dans ces prochaines pages et permettront de mieux comprendre comment a été géré et organisé la réalisation. Cela permettra aussi à (aux) personne(s) qui reprendra(ont) le site web de gagner du temps pour la suite sachant qu'il y a plusieurs améliorations possibles à faire afin de rendre le site web encore plus interactif.

6.2. Sprint 1

6.2.1. Tâches planifiées

Quatre tâches avaient été planifiées dans ce premier sprint :

1. avoir un site sobre et simple – web (5)
2. avoir un site sobre et simple – mobile (5)
3. naviguer entre les pages (1)
4. créer un nouveau diagramme (3)

Mais avant de pouvoir créer un nouveau diagramme ou de n'avoir ne serait-ce qu'une ébauche de site, la première chose à faire a été de choisir les outils de développements, la librairie graphique à utiliser et le langage de développement qui sera utilisé par la suite. Ces trois tâches auraient dû figurer dans le Product Backlog mais au moment de sa rédaction elles ont été oubliées alors qu'elles sont pourtant très importantes puisque toute la suite du projet dépendra de la réalisation de ces tâches. Par contre intuitivement elles avaient été comptées et c'est pourquoi le nombre de « Story Points » planifié, 14, était moins important que lors des sprints suivants.

6.2.2. Choix des outils de développement

Dans un précédent projet de développement web, NetBeans² avait été utilisé pour le développement. Puisque personne d'autre ne travaillait sur le projet, un serveur web local suffisait. Comme une base de données devrait être intégrée au site pour la gestion des utilisateurs et des diagrammes, c'est Easy PHP³ qui a été choisi.

² <http://fr.netbeans.org/>

³ <http://www.easyphp.org/>

NetBeans se présente comme ceci :

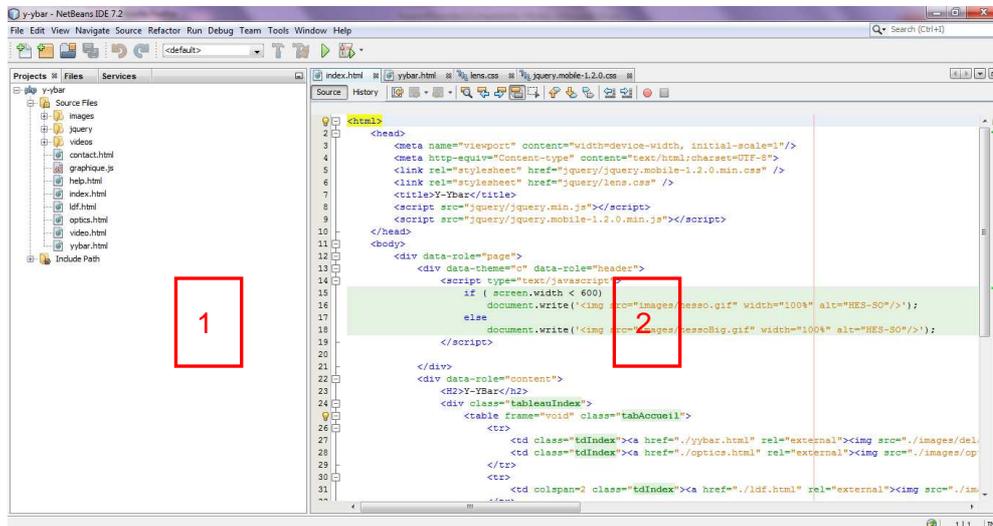


Figure 8 - NetBeans

Le programme est découpé en deux. La partie de gauche est réservée pour la structure du projet avec tous les fichiers qu'il contient. Tandis que la partie de droite permet de travailler sur un fichier en particulier.

Grâce aux différentes couleurs du code et une détection des erreurs, le développeur peut travailler de la même manière que pour un projet .NET ou Java.

La première chose à faire est de créer un projet, en l'occurrence un projet PHP. Puis de choisir le nom du projet, son emplacement, la version de PHP et l'encodage par défaut :

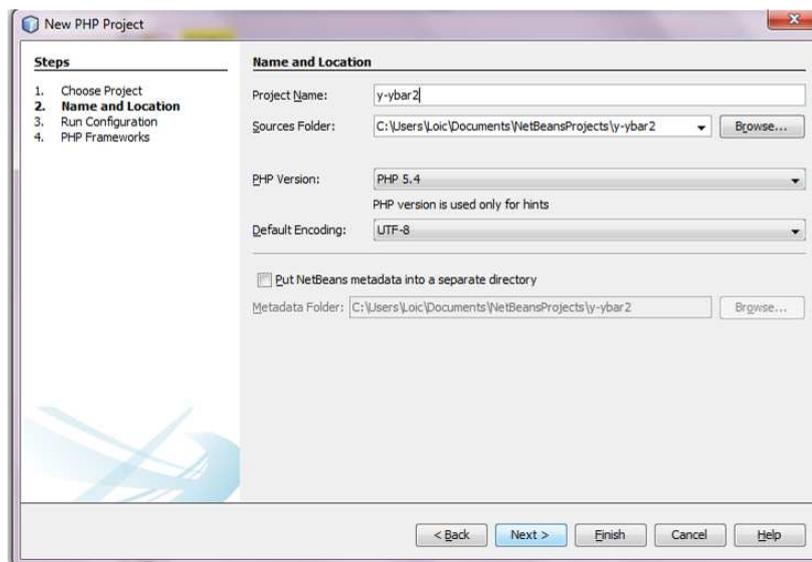


Figure 9 - Création de projet

Une fois le projet créé, un alias doit être créé avec EasyPHP afin de pouvoir un serveur web fonctionnel.

Easy PHP doit être démarré :

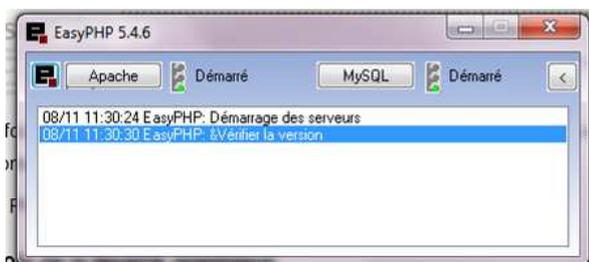


Figure 10 – EasyPHP

Ensuite dans le panneau d'administration qui se trouve à l'adresse <http://127.0.0.1/home/index.php> la création d'un alias est possible et se fait de la manière suivante :



Figure 11 - Création d'un alias

L'alias sera ensuite créé et testable en rentrant l'adresse <http://127.0.0.1/y-ybar/> qui redirigera automatiquement sur la page d'index du projet NetBeans créé précédemment.

Après la mise en place de ces deux outils et donc de l'espace de travail, le développement du projet a pu commencer. Cette solution est satisfaisante dès le moment qu'il n'y a qu'une seule personne qui travail sur le projet et sur un poste de travail unique. Si plusieurs personnes devaient travailler simultanément sur le projet, une solution de type Google Project⁴ aurait pu être choisie par exemple.

Durée : 2 heures

⁴ <http://code.google.com/hosting/>

6.2.3. Choix de la librairie graphique

Après un rapide coup d'œil sur Internet, trois possibilités pour la gestion de graphique « web » sont apparues. :

- HTML 5 avec la balise <canvas>
- HTML avec SVG
- PHP avec une librairie

Après des recherches un peu plus poussées, malgré de bons résultats, HTML 5 a été écarté car cela convient mieux à de la manipulation de pixel (traitement d'images) tandis qu'il est conseillé⁵ d'utiliser SVG pour des graphiques interactif.

SVG (pour Scalable Vector Graphics) permet d'afficher des objets graphiques vectoriels. Les objets sont décrits dans un langage XML qui vont être traité comme les autres éléments HTML. Ainsi, l'interaction avec les éléments créés par SVG se fait de la même façon qu'avec les autres éléments HTML. (Évènements ou application de CSS) Un ensemble de formes primitives (rectangle, cercle, ellipse, etc.) peut être dessiné ou alors via des lignes appelées aussi « Paths ». Il est également possible d'effectuer des transformations, translations ou d'afficher du texte. Ce qui convient parfaitement au projet.

Comme une partie du code, accès à la base de données, se fera en PHP, c'est dans cette direction que c'est ensuite porté les recherches. Au regard des différentes librairies existantes, ChartDirector⁶ semblait convenir parfaitement aux besoins du projet.

Des tests ont ensuite été effectués mais le rendu n'était pas optimal et la complexité de la librairie aurait demandé un temps de prise en main trop importante vu la durée du projet.

De nouvelles recherches ont commencé en partant sur SVG et c'est là que la librairie RaphaelJS a été trouvée. Il s'agit d'une librairie javascript qui simplifie l'utilisation de SVG et le travail avec des graphiques. Après un rapide test de création d'image et une prise en main facile, la décision de valider cette solution a été prise. Ce d'autant qu'une documentation technique précise était disponible tout comme des exemples open source.

Durée : 7 heures

⁵ <http://blogs.msdn.com/b/davrous/archive/2011/05/20/introduction-aux-apis-graphiques-d-html5-svg-amp-canvas-2-4.aspx>

⁶ <http://www.advsofteng.com/>

6.2.4. Choix du langage de développement

Suite au choix d'utilisation de la librairie RaphaelJS, la suite du développement n'allait finalement pas se faire en PHP comme initialement prévu mais en Javascript. Cela permettra malgré tout d'intégrer par la suite du PHP pour la connexion au site par exemple.

Durée : - (temps pris sur le choix de la librairie graphique)

6.2.5. Création du site web

Le site web étant développé avant tout pour les tablettes, l'utilisation de JQuery Mobile⁷ était indispensable. JQuery Mobile est un framework permettant un affichage d'un site Internet optimisé pour les Smartphones et tablettes. Grâce à un fichier javascript et un style CSS, la création d'une page est simplifiée et s'adapte automatiquement suivant la taille de l'écran de l'utilisateur.

Un design approximatif a été créé avec des zones pour les deux graphiques :

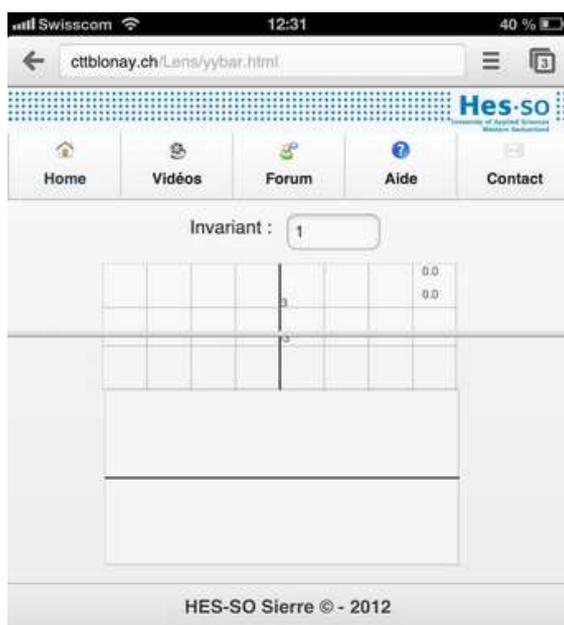


Figure 12 - Design Mobile

Le temps passé sur le design ne devait pas être trop important car M. Stéphane Geiser, qui travaille dans le domaine du graphisme, donnera ses conseils lors du prochain sprint.

La page a été découpée en trois parties. Le haut est réservé pour le menu et la bannière qui pourra être changée ultérieurement. Le contenu est au centre de la page avec les options utilisateurs au début suivi du diagramme de Delano et de la vue de Gauss. Le pied de page est uniquement textuel et reste affiché.

⁷ <http://jquerymobile.com/>

L'affichage web est similaire sauf que la vue de Gauss se retrouve à droite du diagramme de Delano afin d'optimiser la place à disposition sur l'écran :

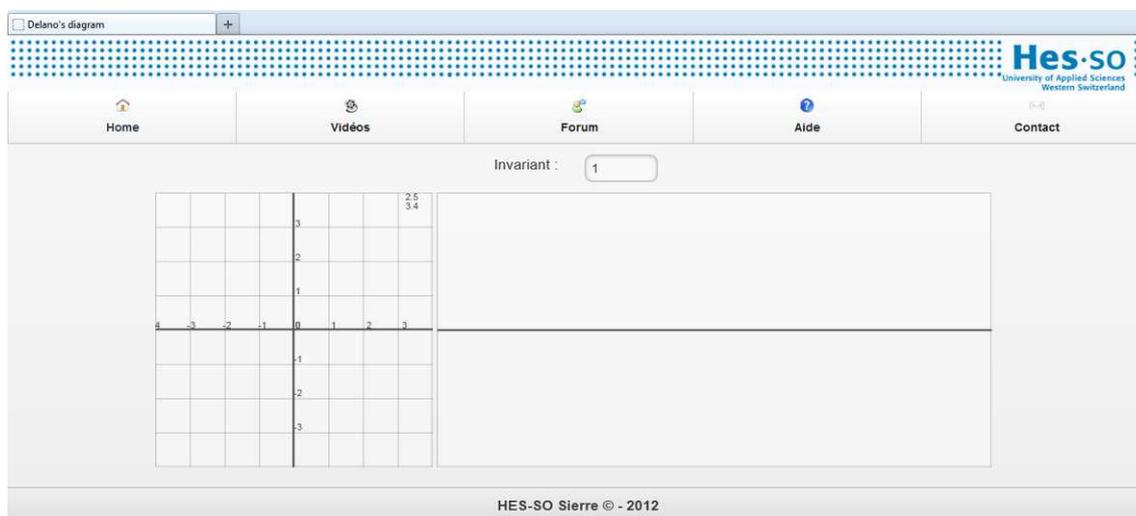


Figure 13 - Design web

Dans les deux cas, l'affichage des valeurs des lentilles et du diagramme se fera à la suite des graphiques et s'adaptera à la taille de l'écran

Durée : 4 heures

6.2.6. Création d'un diagramme

La première chose à faire est de tester si le navigateur utilisé est Internet Explorer. En effet, des problèmes d'affichages sont apparus avec ce dernier donc l'utilisation de Google Chrome ou Mozilla Firefox est souhaitée. Pour ce faire, la récupération des infos du navigateur doit se faire avec un test sur la chaîne récupérée de cette façon : $if(navigator.userAgent.indexOf('MSIE') \neq -1)$. Si les caractères 'MSIE' correspondant à Internet Explorer ne sont pas trouvés, l'index renverra -1 tandis qu'il renverra un chiffre égal ou supérieur à 0 s'il le trouve. (Ce chiffre correspondra à sa position dans le texte)

Ensuite intervient la création d'une <div> où l'affichage du futur diagramme se fera et

```

if ( screen.width < 600){
  dimBase = 304;
  graph = Raphael("graph", dimBase, (dimBase/2));
}
else{
  dimBase = 608;
  graph = Raphael("paper", dimBase, (dimBase/2));
}
  
```

lui assigner un id afin qu'elle soit identifiable par la suite. En l'occurrence comme deux affichages différents sont bien distincts, un test sur la largeur de l'écran client doit être fait. Si la largeur est inférieure à 600 pixels

Figure 14 - Test largeur écran

(correspondant à une petite tablette) alors deux <div> sont créées tandis que pour un affichage « normal » une seule <div> est créée afin de pouvoir afficher les deux graphiques côte à côte. La fonction *Raphael* va créer un espace de travail d'une hauteur deux fois plus

petite que la largeur dans la <div> « graph » ou « paper » qui dépend de la taille de l'écran.

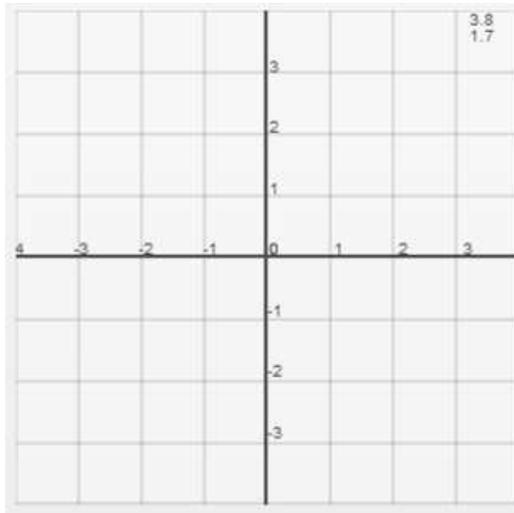


Figure 15 - Création du diagramme

La prochaine étape a été de créer un carré quadrillé avec une légende pour les axes et pour la position de la souris. Grâce à RaphaelJS cela est créé rapidement avec les fonctions : path, rect, text. Path étant pour des lignes, rect pour les rectangles et text pour l'affichage de textes bruts.

Le diagramme est maintenant créé avec un quadrillage qui permettra à l'utilisateur de s'y retrouver facilement.

Durée : 18 heures

6.2.7. Ajout d'une lentille interactive

Grâce aux événements *mouseup* et *touchstart*, la détection du clic de la souris et de la pression du doigt permet de rendre le diagramme interactif. Cela déclenchera une fonction qui créera une ellipse et l'ajoutera dans un tableau d'ellipse. Pour chaque ellipse les propriétés x et y lui sont données, un titre et des valeurs F et F# qui seront expliquées plus tard dans ce document sont également définies. Afin de rendre ces ellipses, appelées lentilles dans notre projet, interactives, l'utilisation du drag et drop est précisé.

Durée : 5 heures

6.2.8. Bilan

Dans ce premier sprint, 4 tâches avaient été planifiées et seulement deux ont pu être validées ainsi qu'une tâche du sprint 2. Le fait de ne pas avoir identifié clairement la mise en place de la station de travail, le choix de la librairie graphique et le choix du langage comme étant des tâches à part entières a été une erreur même si le temps pour leur réalisation avait été planifié.

En ce qui concerne le design du site, j'ai décidé d'attendre le Sprint 2 et la venue de M. Stéphane Geiser pour le faire valider et l'améliorer afin d'éviter d'effectuer le travail à double et perdre du temps. Cela m'a permis de pouvoir prendre de l'avance dans le diagramme de Delano et commencer à le rendre interactif avec des lentilles qui sont créées. Il est également possible de les déplacer tant sur la version ordinateur que sur la version mobile. Une connexion entre ces lentilles est faite mais pas encore validée puisqu'il faudra attendre la suppression d'une lentille pour savoir si elle s'adapte.

En termes de bilan personnel, ce premier sprint a été très intéressant car il s'agissait de ma première expérience sur le traitement de graphiques. Le fait d'avoir obtenu des résultats visuels rapidement a aussi été important que ce soit pour la motivation ou pour le client qui pouvait voir suivre l'évolution du programme.

Le découpage des tâches du Product Backlog m'a également permis de savoir exactement où avancer au moment où la décision de repousser la gestion du design a été prise.

6.2.9. Tâches prioritaires pour le sprint suivant

Par rapport aux tâches planifiées, les tâches suivantes seront à faire le jeudi 25 octobre :

1. avoir un site sobre et simple – web (5)
2. avoir un site sobre et simple – mobile (5)

Tandis que la tâche « ajouter une lentille (3) » du Sprint 2 a déjà pu être validée.

6.3. Sprint 2

6.3.1. Tâches planifiées

Huit tâches ont été planifiées pour ce deuxième sprint :

1. avoir un site sobre et simple – web (5)
2. avoir un site sobre et simple – mobile (5)
3. supprimer une lentille (3)
4. déplacer une lentille (3)
5. avoir un aperçu graphique (10)
6. sauvegarder mon diagramme (2)
7. ouvrir un diagramme existant (2)
8. supprimer un diagramme (2)

Les tâches 1 et 2 seront faites et validées le jeudi 25 octobre. La tâche 4 a été commencée et fonctionne mais ne sera validée qu'au moment où la « zone rouge » sera implémentée.

6.3.2. Suppression d'une lentille

Le temps pour le développement de la suppression d'une lentille a été sous-estimé car cela c'est avéré plus complexe à cause de la connexion qui doit être refaite entre les

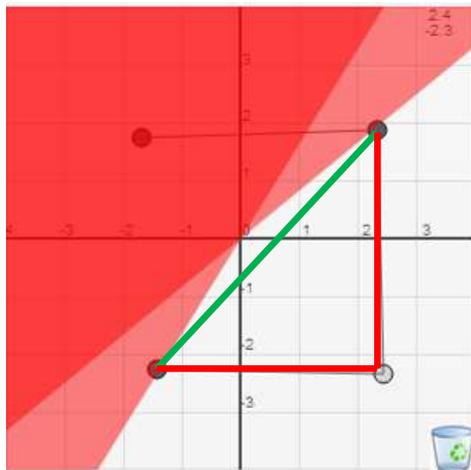


Figure 16 - Suppression lentille

lentilles. La suppression de la dernière lentille a été très rapidement implémentée mais la suppression de la première lentille et d'une lentille centrale a pris plus de temps.

Sur cette image, on voit que la 3^e lentille a changé de couleur et qu'une corbeille est apparue. Cette dernière prend un aspect différent quand elle

sera survolée par la souris : 

Ce changement permettra à l'utilisateur de savoir quand il pourra lâcher la souris et que la lentille sera

en effet supprimée.

Sur l'image de la page précédente, la suppression de cette 3e lentille, qui est connectée aux lentilles adjacentes avec deux traits rouge, oblige une modification des connexions. En effet, les connexions en rouge seront supprimées tandis que la connexion verte sera créée.

La suppression a été découpée en plusieurs parties car cela demande une adaptation du code au moment où la lentille est sélectionnée par l'utilisateur, quand la lentille survole la poubelle et au moment où la lentille est relâchée par l'utilisateur. Soit dans les 3 parties du Drag & Drop.

Dans la première partie du Drag & Drop, appelée « start », la seule chose à implémenter est de faire apparaître une corbeille : `imgBin = paper.image("images/bin.png", 275, 275, 30, 30);` Elle apparaîtra en bas à droite (275;275) du diagramme avec une taille de 30 px sur 30 px.

Dans la partie « move » correspondante au déplacement d'une lentille, un test sur la position du curseur est fait afin de savoir si la corbeille vide est survolée :

```
if (x>=3.3 && y <=-3.2){ imgBin.node.href.baseVal= "images/binFull.png"; }
```

Grâce à cette ligne, la source de l'image va changer et la corbeille pleine apparaîtra. A l'inverse lorsque l'on ne survole pas la corbeille, celle-ci reste vide, il faut donc faire attention à le préciser en changeant également la source de l'image.

Dans la partie « up » (fin du clic de la souris), un test afin de savoir si la corbeille a l'aspect « pleine » est fait et déclenche alors la procédure de suppression.

Cette procédure va dans un premier temps tester si la lentille a des lentilles adjacentes ce qui signifie des connexions qu'il faudra également supprimer. Ensuite, si la lentille est centrale, une nouvelle connexion est créée entre la lentille précédente et la lentille suivante. La lentille est supprimée du diagramme grâce à l'instruction `remove()` puis le tableau de lentille est décalé afin de rechercher rapidement les lentilles adjacentes par la suite. La

procédure est terminée par le calcul de toutes les données. Une vérification est faite et si une erreur est trouvée, un message apparaîtra à l'utilisateur.



Figure 17 - Message d'erreur

Durée : 27 heures

6.3.3. Connexion entre les lentilles

Que ce soit lors de la création d'une lentille, son déplacement ou sa suppression une connexion est affectée par ces actions dès le moment qu'il y a plus d'une lentille sur le diagramme. Grâce à la librairie Raphaeljs et l'exemple trouvé⁸ cela s'est avéré rapide à mettre en place. Seul le type de ligne a dû être modifié afin de n'avoir que des lignes droites et non plus des courbes.

Ensuite en cherchant un peu, une procédure de suppression de connexion avait également déjà été faite⁹.

Après que les deux procédures d'ajout et suppression de connexion ont été implémentées, seule la mise à jour des connexions devait être faite. 3 lignes sont nécessaires à cette mise à jour :

- 1) `paper.connection(connections[i]);`
- 2) `paper.removeConnection(c[cTemp], c[cTempNext]);`
- 3) `connections.push(paper.connection(c[cTempPrev], c[cTempNext], "#5a5e6b"));`

La première ligne met à jour une connexion lorsqu'une lentille est déplacée. La deuxième supprime une connexion entre deux lentilles. (Dans l'exemple c'est entre la lentille supprimée et la lentille suivante). Enfin, la dernière ligne va créer une connexion entre la lentille qui précède celle supprimée et la lentille qui suit la lentille supprimée. On précisera encore la couleur de la connexion.

Durée : 9 heures

6.3.4. Déplacement d'une lentille

Le déplacement d'une lentille se fait grâce au Drag & Drop avec trois fonctions appelées à chaque phase. La première phase est lorsque l'utilisateur clic sur la lentille. La deuxième phase est lorsque l'utilisateur déplace la souris tout en maintenant appuyé le clic. Pour terminer, la troisième phase, lorsque l'utilisateur relâche la souris. Ces trois phases ont des tâches bien précises.

⁸ <http://jsfiddle.net/CHUrB/>

⁹ <http://stackoverflow.com/questions/5599833/adding-custom-remove-method-to-raphael-js-graffle-connection>

Voici la tâche principale de ces trois phases :

Phase 1) Calculer la zone rouge (voir 6.3.5)

Phase 2) Mettre à jour le diagramme en temps réel ; déplacement de la lentille et ses connexions (voir **Error! Reference source not found.**)

Phase 3) Mettre à jour les données et test de suppression (voir 6.3.2)

A la fin de cette partie, le diagramme est fonctionnel et permet l'ajout de lentilles, leur déplacement et leur suppression. La prochaine étape est l'implémentation des calculs afin que le diagramme se transforme en diagramme de Delano...

Durée : 9 heures

6.3.5. « Zone rouge »

Cette zone va limiter les déplacements d'une lentille afin que le système soit réaliste. Il s'agit de la partie la plus technique du projet c'est pourquoi elle est détaillée en deux parties. Une partie explicative de cette zone puis comment elle a été implémentée.

Afin de comprendre un peu mieux ce qu'est cette zone rouge, voici un exemple de l'ancien programme où l'utilisateur déplace la lentille n°3.

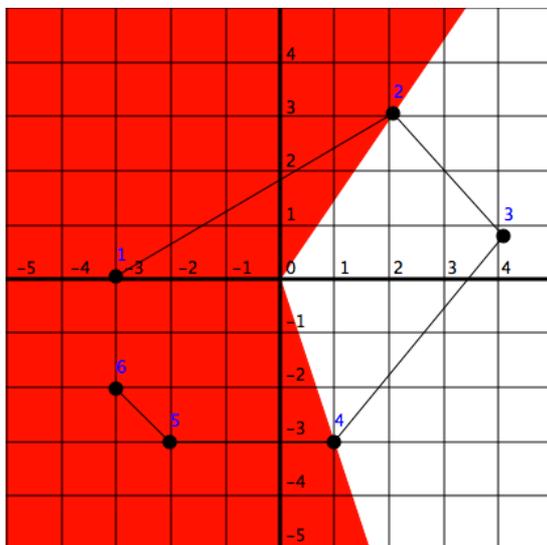


Figure 18 - Zone rouge

Dans ces prochaines pages, une explication de « comment est dessinée cette zone rouge de manière théorique sans aucun calcul » est donnée.

Pour définir cette zone, une droite doit être dessinée. Cette droite doit passer par le point $\{0;0\}$ ainsi que par la lentille précédente. Une deuxième droite est dessinée en passant par la point $\{0;0\}$ et le point suivant ce qui donne ceci comme résultat :

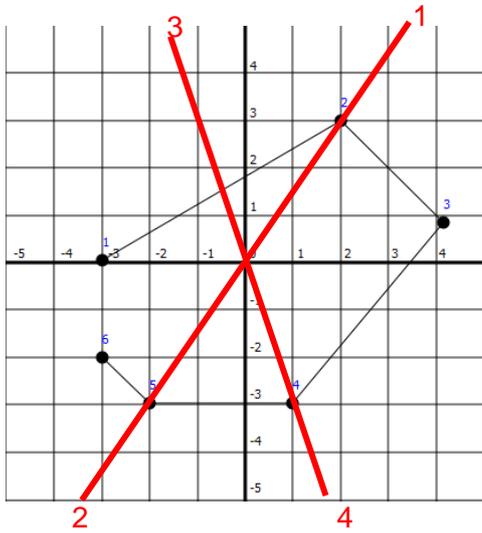


Figure 19 - Zone rouge calculs (1)

Comme l'exemple ci-dessus, les deux droites vont d'un bord à l'autre ce qui donnera 4 points au bord du graphique.

Pour terminer la définition, la zone la plus restrictive est prise. Soit les points 1 et 4 dans notre exemple :

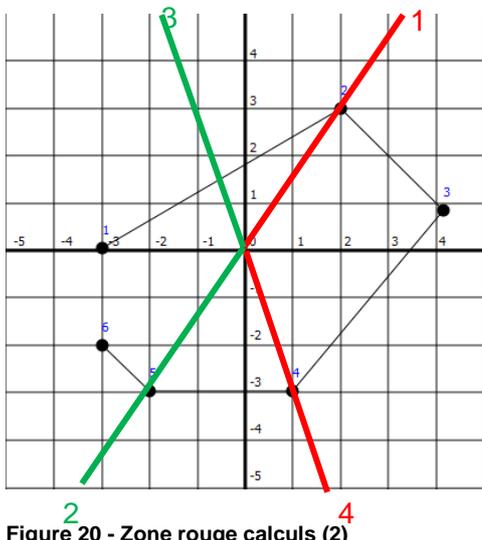


Figure 20 - Zone rouge calculs (2)

Maintenant que les droites restrictives ont été déterminées, le polygone peut être dessiné :

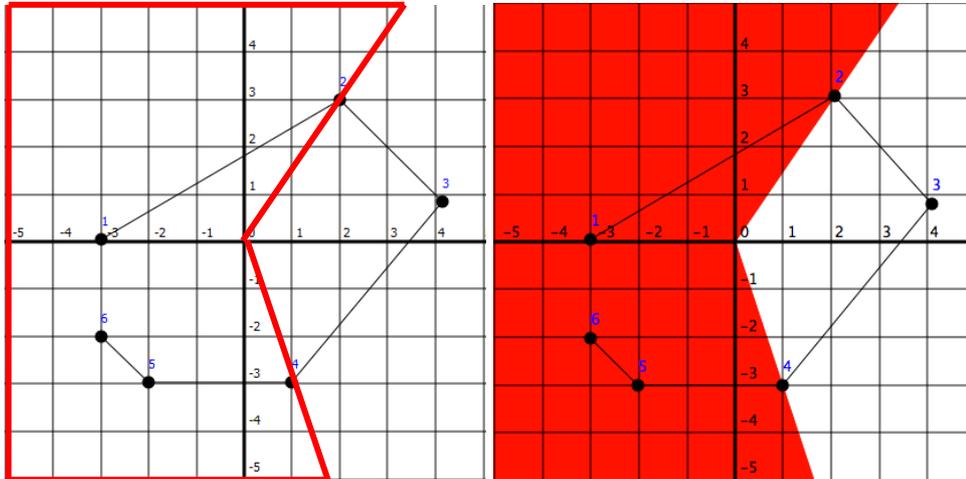


Figure 21 - Zone rouge polygone

Cette comparaison entre le polygone et le résultat souhaité (première image du chapitre) permet de constater que le polygone est correct.

Cette explication théorique a été la première chose à comprendre et a déjà pris pas mal de temps puisqu'aucune documentation à ce sujet n'a pu être trouvée sur Internet et qu'il était très difficile de comprendre le code de l'ancien programme.

Au niveau de l'implémentation, le piège était de dessiner le polygone comme sur la Figure 21 - Zone rouge polygone. Pourquoi le piège ? Du fait que dans l'exemple choisi, une lentille centrale a été choisie et donc deux droites ont été dessinées. Tandis que sur une lentille extérieure, une seule droite est dessinée ce qui change passablement les calculs des coordonnées du polygone et donc deux procédures auraient dû être créées pour le même but souhaité.

L'astuce pour n'avoir qu'une seule procédure est de dessiner un polygone pour chaque lentille adjacente et donc de ne plus se soucier de la zone la plus restrictive. C'est pour cela que sur le site web (Figure 16 - Suppression lentille par exemple) une zone est d'un rouge plus opaque. C'est parce que deux polygones sont superposés à cet endroit là.

Au niveau du code à proprement parlé, la première chose à faire a été de déterminer les coordonnées du polygone. La principale difficulté était de savoir où la droite rencontrait le bord du graphique, donc de calculer la droite passant par la lentille et $\{0;0\}$. En suivant les explications d'un pdf¹⁰, les calculs ont été mis en place. Deux fonctions (variationAbscisse et variationOrdonnee) ont été créées et permettent de récupérer les coordonnées $\{x;y\}$ extrêmes de la droite qui sont ensuite converties en coordonnées image.

Les autres points du polygone étant aux angles du diagramme, ils étaient identifiables rapidement. Il a fallu malgré tout savoir dans quel sens dessiner le polygone afin d'éviter de dessiner la zone censée être blanche. Il a également fallu récupérer les bords sur lesquels se trouvaient les coordonnées de la droite afin de ne pas dépasser ces points.

Une fois ces points récupérés, le dessin du polygone a pu être fait. Pour ce faire, un « path » (droite) a été créé avec plusieurs coordonnées et une couleur de fond : `rouge.attr({stroke: '#FF0000', 'stroke-width': 1, fill: '#FF0000', opacity: 0.5})`

La couleur des bords est la même que celle du fond avec une opacité à 50%. Ensuite il faut modifier l'attribut path de la façon suivante : `rouge.attr({path: coords});` Pour mieux comprendre comment cela fonctionne, un exemple est disponible à cette adresse : <http://davidbau.com/javascript/learn/10-polygon.html#>

Pour terminer il a fallu empêcher l'utilisateur de survoler la zone avec une lentille. Afin de gagner du temps d'exécution et de devoir recalculer les polygones à chaque fois, la solution la plus simple a été de modifier le curseur au moment où l'utilisateur rentre ou sort de la zone. Un test du type de curseur est fait au moment du Drag & Drop lorsque l'utilisateur déplace la souris. Si le curseur est une main alors il va déplacer la lentille sinon il ne la déplace pas. Cette solution ne fonctionne évidemment pas pour une version mobile puisqu'il n'y a pas de curseur. Donc pour cette solution, il est nécessaire de refaire les calculs à chaque déplacement.

Ces calculs vont déterminer si la position du curseur se trouve dans un polygone ou non. Ils seront également utilisés lorsque l'utilisateur va vouloir rajouter une lentille. S'il essaye de rajouter une lentille dans une zone non-réaliste alors il ne se passera rien.

Durée : 27 heures

¹⁰ http://jl.maths-lfb.fr/seconde/reperage/methode_droites.pdf

6.3.6. Affichage des valeurs du diagramme

La dernière étape de base pour avoir un diagramme de Delano utilisable était l'affichage de toutes les valeurs comme la longueur focale. Après discussion avec le client, il a été décidé de n'afficher que les valeurs suivantes :

- Position x et y
- Diamètre
- Longueur focale (F)
- F#
- Distance entre les lentilles

Pour récupérer chaque valeur des fonctions ont été créées. Elles ont été reprises du code source de l'ancien programme afin d'éviter toute erreur de calcul.

La position x et y étant propre à chaque lentille sans interaction avec une lentille adjacente, ces données sont enregistrées directement sur la lentille.

```
c[nbPoints] = paper.ellipse(clickevent.layerX,clickevent.layerY,r,r)
  .attr({"title":nbPoints,"fill":'#5a5e6b', "cursor":"pointer"})
  .drag(move, start,up)
  .data("x",x)
  .data("y",y)
  .data("f","")
  .data("f#", "");
```

Figure 22 - Code création d'une lentille

On constate sur cette image qu'on initialise également f et f# qui seront utilisées pour la suite, lorsque la lentille devient centrale.

Pour l'affichage, une fonction spéciale a été créée et permet d'afficher les données dans une <div> se trouvant après les graphiques. C'est avec l'instruction : `document.getElementById("tabDonnees").innerHTML` qu'est défini le contenu de la <div> tabDonnees.

Dernier paramètre pris en compte, la couleur de chaque ligne change suivant la valeur de F#. Si F# est plus petit que 5 alors la ligne sera rouge, entre 5 et 10 rosée, entre 10 et 15 orange et plus grand que 15 verte.

Durée : 18 heures

6.3.7. Design définitif du site web

Après une discussion avec M. Stéphane Geiser, il ressort que le design du site était

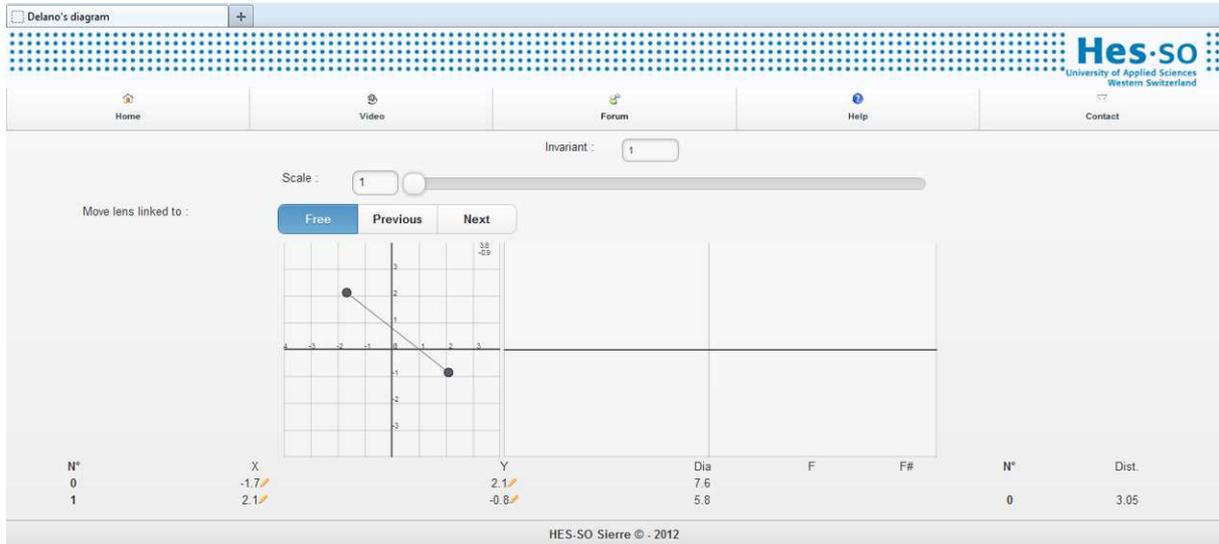


Figure 23 - Design standard

suffisamment sobre et simple. Mais quelques conseils ont été donnés et appliqués :

- Centrer les données
- Utiliser le maximum l'écran donc graphiques côte à côte en cas d'utilisation standard
- Réduire la bannière en hauteur
- Données dans un seul tableau
- Quadrillage du diagramme plus clair afin de mettre en valeur les éléments
- Lentilles un peu plus grosse sur la version mobile

Ces remarques font ressortir le contenu principal à savoir le diagramme Delano et la vue de Gauss. Voici les deux affichages (standard et mobile) sans la vue de Gauss.

Durée : 4 heures

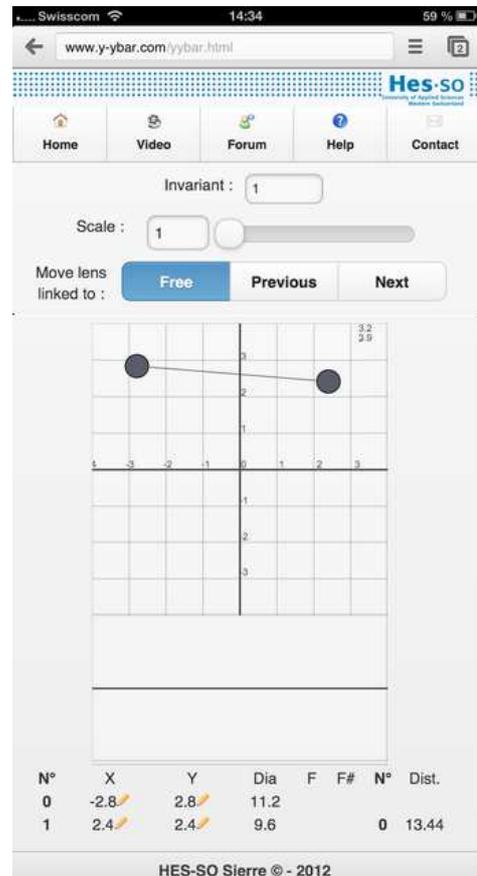


Figure 24 - Design mobile iPhone

6.3.8. Impression du diagramme et de ces valeurs

En ce qui concerne l'impression, il a été décidé d'utiliser l'impression du navigateur web par défaut. Comme le site est sobre et n'affiche que le contenu utile (pas de publicité par exemple) cela permet à l'utilisateur d'avoir des impressions lisibles.

Durée : 1 heure

6.3.9. Bilan

Le bilan de ce sprint est bon puisque le diagramme de Delano fonctionne et que le site a un design validé par une personne travaillant dans le graphisme.

Sur les huit tâches prévues, trois n'ont pas été faites. En effet, la sauvegarde, l'ouverture et la suppression de diagramme demandant une base de données et un compte utilisateur a été mis de côté pour l'instant afin d'avoir avant tout un site web fonctionnel et utilisable. Le déplacement d'une lentille ayant pris beaucoup plus de temps que prévu à cause de la zone rouge, la redéfinition des priorités a dû être faite et c'est pour cela que la connexion à une base de données est redescendue en priorité 2.

Après deux semaines de développements. Je me rends compte que la durée des tâches a mal été évaluée et que de ce fait les graphiques « Project Velocity » et « Release roadmap » qui seront présentés à la fin du sprint 3 ne seront pas forcément représentatifs du travail accomplis.

Pour la dernière semaine de développement, un point avec le client sera fait mardi 30 octobre. Cela permettra de définir de nouvelles priorités qui seront mieux évaluées en termes de temps de travail connaissant mieux le sujet.

Avant cela, la vue de Gauss devra être implémentée et le site mis en ligne afin de pouvoir demander l'avis de M. Logean sur le programme.

6.3.10. Tâches prioritaires pour le prochain sprint

Comme les tâches ont été et seront redéfinies, difficile de s'appuyer sur le Sprint Backlog mais il faudra avoir un produit fonctionnelle est utilisable en fin de sprint. Il sera donc primordial de pouvoir modifier l'invariant (3) et d'avoir une vue graphique (10).

Plus de détails seront donnés au chapitre 38

6.4. Sprint 3

6.4.1. Tâches planifiées

Deux tâches sont planifiées pour le premier jour du sprint avant un point de situation :

- 1) Aperçu graphique – vue de Gauss (10)
- 2) Modification de l'invariant (3)

L'aperçu graphique vaut 10 Story Points puisque le diagramme de Delano a pris 2 semaines. Cette tâche devrait quand même prendre moins de temps car il n'y a plus de prise en main de la librairie RaphaelJS à faire et que le graphique est statique.

La modification de l'invariant influence certains calculs et ne devrait pas prendre trop de temps à implémenter puisque ces calculs avaient été préparés à cette modification.

6.4.2. Aperçu graphique – vue de Gauss

L'aperçu graphique permet à l'utilisateur d'avoir une image de son système optique :

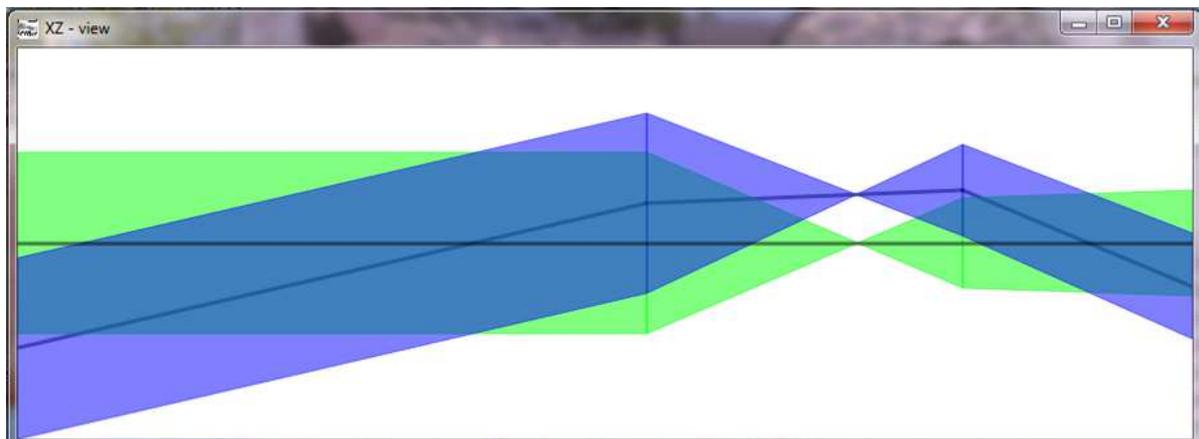


Figure 25 - Vue de Gauss, ancien programme

Deux courbes sont représentées :

- Rayon marginal en vert (marginal ray)
- Rayon principal en bleu (chief ray)

Comme il s'agit d'un graphique statique, aucune interactivité avec l'utilisateur, il doit être généré à chaque modification du diagramme de Delano. Comme pour la zone rouge, il faut dessiner un « path » à plusieurs coordonnées.

Une fonction spéciale a été créée pour dessiner ce graphique (`drawGraph`). Le premier test à faire est de vérifier qu'il y a bien au minimum 3 lentilles sinon ce n'est pas un système optique complet. Une fois ce test passé on peut définir les deux graphiques.

Ensuite il faut trouver la distance totale en faisant la somme de toutes les distances des connexions. Puis il faut définir les points les plus éloignés ce qui définira la hauteur du graphique afin qu'il adapte sa taille et ne reste pas petit si les valeurs sont basses.

Maintenant que la différence de hauteur et la longueur est connue, il est possible de définir ce que chaque point vaut comme pixels afin d'optimiser au mieux la place qu'il y a à disposition.

L'avant dernier point est de définir les coordonnées de chaque graphique pour que le *Path* se remplisse et donne l'aspect souhaité.

La dernière chose à faire était d'afficher des traits pour que l'utilisateur puisse voir rapidement où se trouvaient ces lentilles. L'échelle a été rajoutée plus tard suite à une remarque de M. Eric Logean.

Au final l'affichage de la vue de Gauss a cette allure :

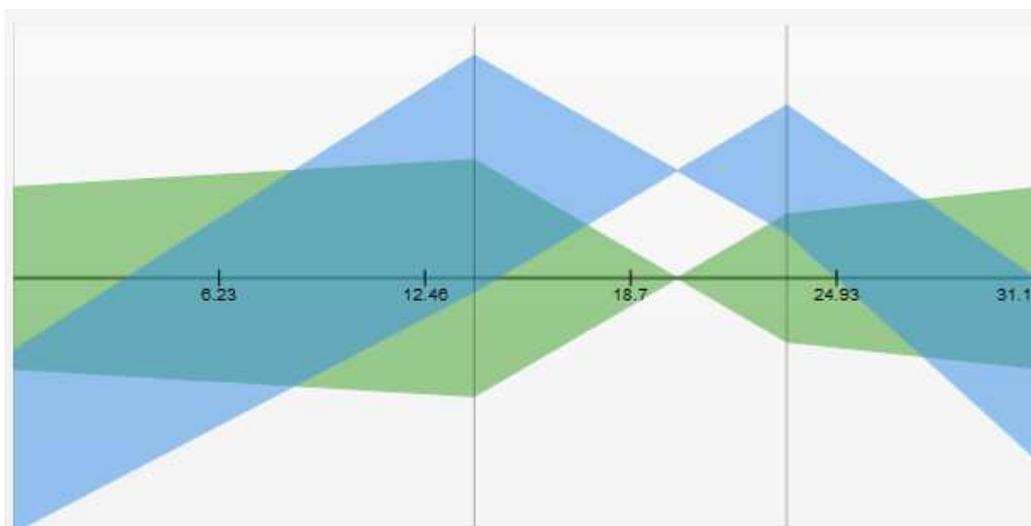


Figure 26- Vue de Gauss, nouveau programme

Durée : 18 heures

6.4.3. Modification de l'invariant

L'invariant influence uniquement les valeurs F et $F\#$ et donc la distance. Cela aura comme répercussion de modifier l'échelle de la vue de Gauss et les valeurs qui sont affichées. Il faut donc mettre à jour l'échelle et mettre à jour les données.

Le plus simple pour l'utilisateur était de mettre un champ où il n'y aurait qu'à modifier la valeur pour que tout se mette à jour automatiquement.

Au niveau du codage, rien de particulier. Grâce aux fonctions créées, il suffisait de les appeler donc cela a demandé très peu de temps pour implémenter cette fonction.

Durée : 3 heures

6.4.4. Point de la situation

Au deuxième jour du troisième sprint, il a été décidé de faire un point de la situation avec M. Geiser et de lister toutes les tâches qui devaient encore être faites puis de leur donner une priorité afin qu'à la fin du sprint, le site web soit déployable et utilisable.

Huit tâches ont été listées suivant l'ordre suivant :

- Avoir un site complet avec une page d'accueil
- Mettre en place une contrainte de déplacements
- Pouvoir définir les valeurs x et y manuellement
- Protéger le code source
- Affichage d'une FAQ
- Affichage d'une vidéo
- Affichage des pupilles sur la vue de Gauss
- Pouvoir exporter les valeurs x et y dans un fichier texte et les réutiliser.

M. Eric Logean a également été contacté afin qu'il teste le site web. Le bilan était bon mais avait trois améliorations possibles :

- Échelle variable pour la vue y-yb. Certains systèmes peuvent demander des points avec des valeurs plus grandes que 4.
- Indicateur d'échelle pour la vue 'classique' (de Gauss) et un petit espace entre les 2 vues.
- Possibilité de lire et d'écrire des fichiers textes avec les données y et yb.

Comme les deux premières propositions sont importantes, il a été décidé de les implémenter juste après la modification des valeurs x et y mais avant la protection du code source. Soit en position 4 et 5.

Durée : 2 heures

6.4.5. Architecture du site web

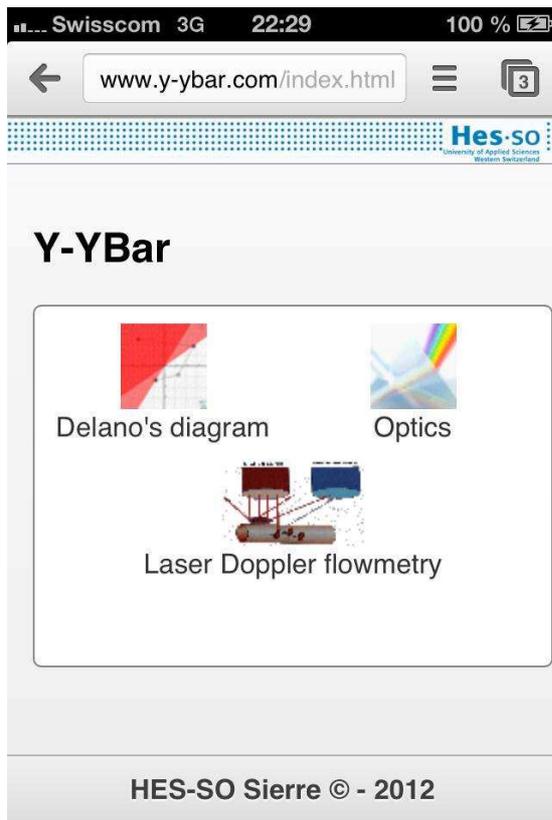


Figure 27 - Page d'accueil

Trois chapitres du site web seront disponibles dans le futur.

- Delano's diagramm
- Optics
- Laser Doppler flowmetry (LDF)

Toujours dans l'optique d'avoir un site avec un design « tablette », des icones ont été préféré plutôt que de longs textes. Cela permet d'avoir un site sobre et le plus simple possible pour l'utilisateur.

Seul la partie du diagramme de Delano est à implémenter, les pages « optics » et « LDF » seront mise à jours ultérieurement par M. Geiser.

Durée : 9 heures

6.4.6. Contrainte de déplacement

Deux contraintes de déplacements ont été implémentées :

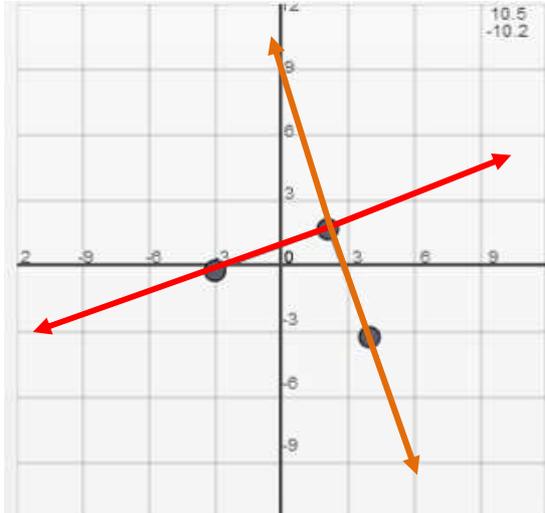
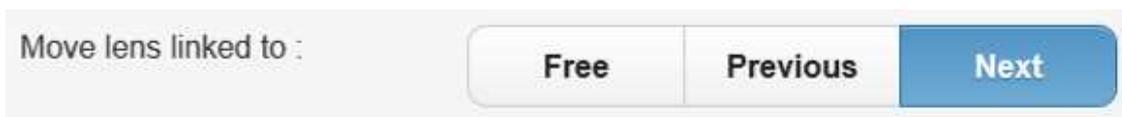


Figure 28 - Contraintes de déplacements

Comme l'indique ces deux flèches, les déplacements de la lentille centrale se font suivant les lentilles adjacentes. La lentille 2 se déplace sur la même droite que la connexion avec la lentille précédente (rouge) ou la lentille suivante (orange).

Difficile de faire que pour l'utilisateur ce soit intuitif et dans l'ancien programme cela était représenté par deux icônes donc pas forcément intuitif non plus. Il a donc été choisi de le faire avec des radios boutons (aspect mobile) :



Par défaut l'option « Free » est sélectionnée et permet un déplacement libre pour l'utilisateur.

Au niveau de l'implémentation, Les fonctions de variation d'ordonnées et d'abscisses utilisées pour le calcul du polygone ont été reprises afin de déterminer la droite. Comme il faut avoir au moins une variation donnée en fonction du curseur, j'ai décidé de varier l'axe des x en fonction de la position y du curseur. Cela signifie que la lentille va se déplacer lorsque l'utilisateur déplacera le curseur en hauteur. S'il reste sur la même hauteur, la lentille ne bougera pas.

Durée : 9 heures

6.4.7. Modification de x et y

La modification de x et y permet à l'utilisateur de faire de légers ajustements et d'avoir une meilleure précision lors de systèmes optiques de pointes.

Le plus rapide à implémenter a été de rajouter un événement aux cellules du tableau de données puis d'afficher un message à l'utilisateur qui lui proposera de modifier la valeur existante. Pour que ce soit clair à l'utilisateur, un crayon a été rajouté à côté de chaque valeur et le curseur se transforme en main lorsqu'il survole une valeur.

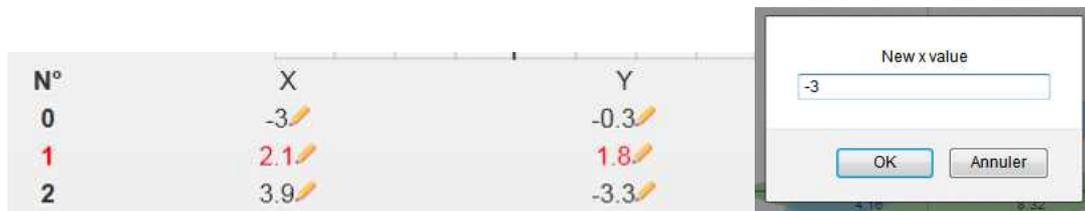


Figure 29 - Modification d'une valeur

Durée : 4 heures

6.4.8. Modification de l'échelle du diagramme

Première proposition de M. Logean, la modification de l'échelle du diagramme de Delano a demandé une adaptation graphique et un recalcul de la position de chaque lentilles.

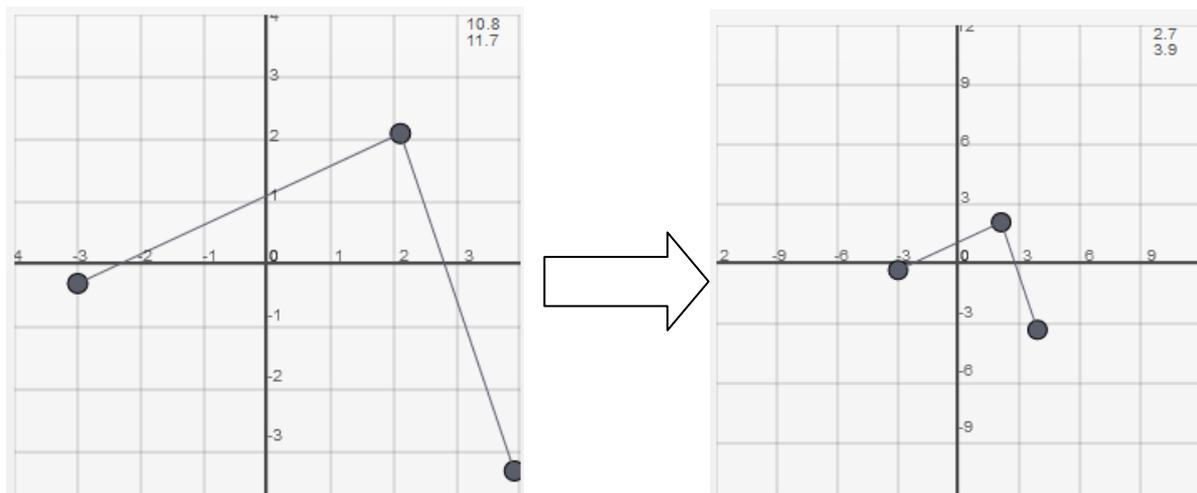


Figure 30 - Modification de l'échelle exemple

Comme le montre cet exemple, l'échelle de l'image de gauche va de -4 à 4 tandis que l'échelle de l'image de droite va de -12 à 12. Soit un coefficient multiplicateur de 3. Cela

permet à l'utilisateur de créer des diagrammes avec un maximum de précision en gardant un aperçu optimisé pour l'utilisation sur tablettes.

En premier il faut supprimer la légende des deux axes et la remplacer par la nouvelle légende suivant le coefficient choisi par l'utilisateur. Ensuite il faut repositionner chaque lentilles. Les valeurs et la vue de Gauss ne sont pas affectées par ce changement il n'y a donc pas besoin de mettre à jour ces affichages.

Durée : 9 heures

6.4.9. Echelle de la vue de Gauss

Deuxième proposition de M. Logean, le rajout d'une échelle pour la vue de Gauss. Pour ce faire, lors de la génération de cette vue, il faut récupérer le distance totale puis la diviser afin d'avoir 5 valeurs qui seront affichées à l'utilisateur.



Figure 31 - Echelle vue de Gauss

Durée : 2 heures

6.4.10. Affichage d'une vidéo

L'affichage d'une vidéo se fait avec la balise HTML 5 <video> où la source est précisée :

```
<video controls src="/videos/demo.webm" type="video/webm" width="300">Demonstration</video>
```



Figure 32 – Vidéo

Durée : 1 heure

6.4.11. Code source protégé

Il est difficile de protéger le code javascript d'un site web car même s'il se trouve dans un fichier annexe, l'utilisateur peut y accéder en affichant la source de la page puis en cliquant sur le lien pointant vers le fichier. Comme il ne me restait que peu de temps, j'ai pris la décision de faire au plus simple et de désactiver le clic droit sur la page du diagramme de Delano. Il est évidemment qu'une personne voulant à tout prix le code du site pourra l'obtenir mais si c'est par simple curiosité, un utilisateur « lambda » ne sait pas forcément afficher le code source sans passer par le clic droit.

Durée : 2 heures

6.4.12. FAQ

Pour l'aide, il a été choisi de partir sur une FAQ avec une liste de questions et réponses qui sera ajoutée par la suite au site web. Le but étant de construire l'architecture de la page.

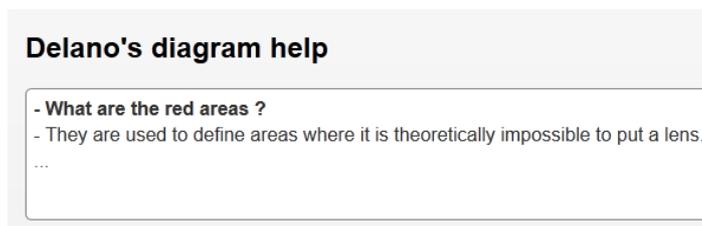


Figure 33 – FAQ

Durée : 1 heure

6.4.13. Déploiement du site web

La dernière chose à faire a été de déployer le site web sur le site www.y-ybar.com

Pour ce faire, M. Geiser m'a fourni les informations ftp puis avec le logiciel Filezilla¹¹ il ne restait plus qu'à mettre les fichiers sur le serveur. Un léger problème de configuration a empêché le déploiement au début mais après avoir contacté Easygiga, l'hébergeur, tout a été résolu et le site a pu être mis en ligne.

Durée : 4 heures

¹¹ <http://filezilla-project.org/>

6.4.14. Bilan

Ce dernier sprint a été intense et le résultat final est celui choisi par le client grâce à la redéfinition des tâches au début du sprint.

Seul l'affichage des pupilles qui demandaient de nouveaux calculs techniques et la possibilité d'enregistrer ces données n'ont pu être faites. Mais ces deux tâches n'empêchent pas l'utilisation du site et seront des améliorations possibles pour le futur.

D'un point de vue gestion du temps, il a été essentiel de faire cette mise au point et d'avoir passé du temps avec le client en début de sprint et non pas en fin de sprint comme prévu. Grâce à cela, les tâches qui étaient les plus importantes pour les utilisateurs futurs ont été implémentées. N'étant pas un spécialiste du domaine optique, je pouvais difficilement me rendre compte s'il était plus important d'avoir une contrainte de déplacement ou que les pupilles soient affichées. (Elles ne l'étaient pas non plus dans le précédent programme) Avec les avis de M. Geiser et M. Logean il m'était plus facile de savoir ce qui était important.

Le quatrième sprint est réservé pour la rédaction de la documentation finale et des corrections minimales de bugs.

Les graphiques *Realease Road map* et *Sprint Velocity* seront détaillés dans le chapitre Retour d'expérience

7. Améliorations possibles

Deux tâches n'ont pas été finies lors du sprint 3. En plus de celles citées dans le Product Backlog.

- Affichage des pupilles sur la vue de Gauss
- Possibilité de se connecter au site via un login/mot de passe
- Enregistrer les données des diagrammes afin de le réutiliser ultérieurement.
- Avoir une console d'administration afin de gérer les utilisateurs, les diagrammes, les vidéos, la FAQ et le forum.
- Implémenter le forum afin d'avoir un site interactif et dynamique. Cela permettra d'avoir une communauté.
- Faire en sorte que le site soit détecté par les navigateurs de recherche et placé en tête lorsqu'une personne rentre « y-ybar », « Delano »,... comme recherche.
- D'autres contraintes de déplacements peuvent être implémentées. (Verticales et horizontales)
- La mise en ligne de plusieurs vidéos serait également une bonne aide pour les utilisateurs.
- La FAQ devra être complétée.
- Maintenant que le site est utilisable, le point important est de le faire connaître tant au niveau des écoles spécialisées que sur les entreprises ayant potentiellement besoin de simuler un système optique. Pour ce faire, une lettre de présentation pourrait être envoyée par exemple.
- Comme dit au début du projet, l'intégration à un livre de M. Schmidt pourrait également permettre de faire connaître le site. Et cela peut servir de base pour expliquer le diagramme de Delano.
- Avant cela il est important que plusieurs utilisateurs testent le site de la même façon que M. Logean afin d'avoir un maximum de retour et pouvoir l'améliorer en fonction des besoins des utilisateurs.

8. Retour d'expérience

Pour le retour d'expérience, je ne vais parler que de la gestion des sprints. Le retour sur le projet dans sa globalité se trouvant dans la Conclusion.

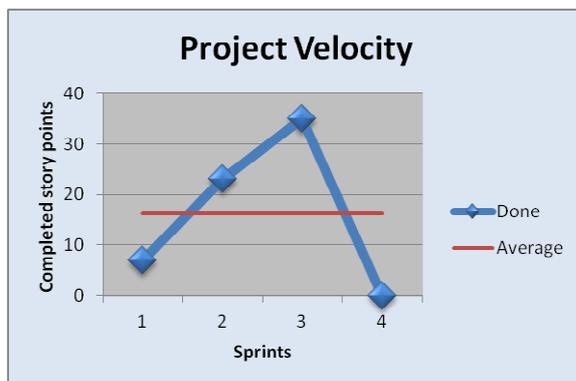


Figure 34 - Project Velocity

Le premier graphique de gestion de projet est le *Project Velocity*. Il montre que plus les tâches ont été planifiées avec précision comme le sprint 3, plus cela était efficace. Il était aussi plus facile de déterminer les tâches à faire en fin de sprint 2 qu'au début du sprint 1. Le sprint 4 initialement planifié a finalement été réservé pour la documentation. Pourquoi ce choix et pas une documentation faite en cours de projet ? Le fait de faire la

documentation en cours de projet risquait de me faire perdre du temps. Passer du code au document Word, avec le temps de se remémorer le travail à faire et l'état actuel aurait été du temps précieux perdu et au cas où à la fin du sprint 3, il y avait encore une-deux tâches à faire, il aurait été possible de déborder un peu sur le rapport final puisque le plus important est que le client soit satisfait du résultat final.

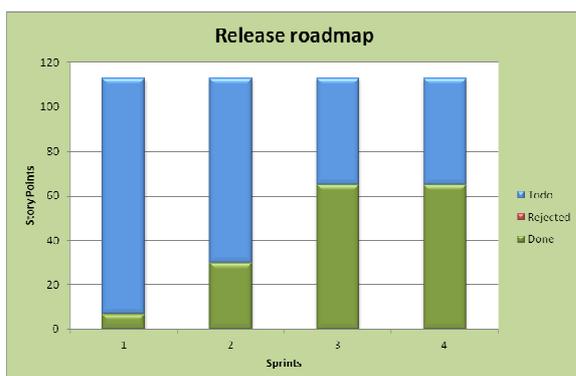


Figure 35 - Release roadmap

Même constat pour le *Release roadmap* où cela est peu significatif puisque les tâches restantes sont de priorité 2 alors que certaines tâches comme le déploiement du site web ou la mise en place d'une station de travail n'ont pas été comptabilisées. Aucune des tâches commencées n'a par contre été refusée. Cela a été possible grâce au fait de passer régulièrement chez le client et d'avoir toujours son avis.

Si cela avait été à refaire, je détaillerais d'avantages les tâches comme le déplacement d'une lentille avec la zone rouge. J'avoue avoir fait abstraction de ces deux graphiques pendant les sprints car le fait d'avoir mis des tâches de priorités 2 faisait que, dès le début, je savais qu'il serait impossible de tout faire et donc difficile d'avoir comme objectif de finir à 100%. Ce d'autant que certaines tâches se sont rajoutées en cours de sprint.

9. Conclusion

Le projet fût très intéressant. Le choix de le faire dans un domaine qui m'était inconnu a été motivant car chaque jour j'apprenais quelque chose. L'analyse du programme existant et celle plus techniques sur le futur programme ont été utiles pour débiter le projet sur de bonnes bases. Par contre il fallait s'adapter au fur et à mesure et rester concentrer sur une seule tâche à la fois.

La gestion du projet de manière agile n'a pas été des plus efficaces. Je pense que cela est surtout utile dans des projets de groupe comme nous avons fait lors de cette dernière année. Etant seul dessus et vu la courte durée, tout le potentiel des méthodes agiles ne pouvait être exploité à mon avis.

Le plus intéressant a sans aucun doute été de réfléchir aux calculs à faire pour les fonctions techniques et surtout la zone rouge. Ayant beaucoup apprécié le cours d'Algorithme, là j'étais servi et pu mettre en pratique certains points vu en cours et c'est toujours intéressant de pouvoir mettre en pratique les choses vu de manière purement théorique.

Le fait de passer une journée par semaine à la HES-SO de Sion, chez le client, fût un excellent choix également. Il aurait été impossible d'avoir un tel résultat sans cela et je pense qu'il est important d'avoir un contact régulier avec le client lors d'un projet dans un domaine aussi technique afin d'éviter de partir sur une mauvaise route, de mauvais calculs ou sur des tâches paraissant prioritaires pour le développeur mais pas forcément pour le client.

Avoir un résultat à chaque fin de sprint a aussi été motivant pour moi mais également utile pour le client qui pouvait se rendre compte de ce qui restait à faire et a pu donner son impression. Ce qui a permis de faire quelques ajustements notamment au début du sprint 3.

La durée à disposition pour faire le projet était bonne mais évidemment plus il y a de temps plus il est possible d'approfondir le programme et peut-être qu'avec une semaine supplémentaire il aurait été possible de l'améliorer encore d'avantages.

Au final, avec un site web qui est fonctionnelle et utilisable et qui a été développé tout en accord avec le client fait que pour moi le projet est réussi même si certaines choses auraient pu être améliorée comme la gestion des tâches avec les méthodes agiles.

10. Références

La majeure partie des sites visités sont en notes de bas de pages dans la section où ils ont été utilisés. Seuls trois sites expliquant les principes de base de l'optique :

Institute of Applied Physics (2012). Design and Correction of optical Systems. Consulté le 20 septembre 2012, disponible sur :

<http://www.iap.uni->

[jena.de/iapmedia/Lecture/Design+and+correction+of+optical+systems1351638000/DaCO_S12_Lecture_Part_5_Properties_of_optical_systems_105.pdf](http://www.iap.uni-jena.de/iapmedia/Lecture/Design+and+correction+of+optical+systems1351638000/DaCO_S12_Lecture_Part_5_Properties_of_optical_systems_105.pdf)

Athénée de Luxembourg (?). Chapitre 4 : Lentilles convergentes. Consulté le 20 septembre 2012, disponible sur :

<http://158.64.21.3/physics/Downloads/Deuxi%C3%A8me/Cours/04%20Lentilles%20convergentes.pdf>

Physique Chimie au collège et au Lycée (?). Les Lentilles convergentes – animation flash. Consulté le 20 septembre 2012, disponible sur :

http://physiquecollege.free.fr/physique_chimie_college_lycee/quatrieme/optique/lentille_convergente.htm