



École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique
5^e année
2012 - 2013

Rapport de projet de fin d'études

**Adaptation des images d'un site web pour
la compensation du daltonisme**

Encadrants

Mohamed Slimane
mohamed.slimane@univ-tours.fr
Sebastien Aupetit
aupetit@univ-tours.fr

Etudiant

Pierre FOURREAU
pierre.fourreau@etu.univ-tours.fr

DI5 2012 - 2013

Université François-Rabelais, Tours

Version du 30 avril 2013

Table des matières

1	Introduction	6
2	L'accessibilité sur le web	7
2.1	Présentation générale	7
2.2	Importance de l'accessibilité	7
2.3	Résolution des problèmes d'accessibilité	7
2.4	W3C et WAI	8
3	La notion d'utilisabilité	9
3.1	Présentation générale	9
3.2	La notion d'ergonomie	9
3.3	Règles de Jacob Nielsen	9
4	Le daltonisme	11
4.1	Présentation	11
4.2	Causes	11
4.3	Types de d'anomalies	11
4.3.1	Monochromatisme	11
4.3.2	Dichromate	12
4.3.3	Trichromate anormal	12
5	Le protocole HTTP	13
6	La notion d'espace colorimétrique	14
6.1	Définition	14
6.2	Le RGB	14
6.3	Le sRGB	15
6.4	CIE Lab	16
6.5	LMS	16
7	Spécifications système	18
7.1	Contexte	18
7.2	Objectifs	18
8	Travail réalisé	19
8.1	Recherches et documentation	19
8.2	Installation de l'environnement de développement	19
8.3	Analyse de l'existant de la plateforme SWAP	19
8.4	Rédaction du cahier de spécifications	19
8.5	Préparation de la présentation orale à mi-parcours	20
8.6	Simuler une vue daltonienne	20
8.6.1	Outils de simulation	20
8.6.2	Manipulation des images	21

8.6.3	Principe de la simulation	21
8.6.4	Fonctionnement de la simulation	21
8.6.5	Interface sur la plateforme SWAP	22
8.6.6	Tests	22
8.6.7	Intérêt	23
8.7	Marquage des images	23
8.7.1	Fonctionnement	23
8.7.2	Intérêt	23
8.8	Mise en cache des images	24
8.8.1	Principe	24
8.8.2	Fonctionnement	24
8.8.3	Intérêt	25
8.8.4	Tests	25
8.9	Correction des images	25
8.9.1	Algorithmes de correction	26
8.9.2	Passage par l'espace de couleur LMS	26
8.9.3	Passage par l'espace de couleur CIE Lab	26
8.9.4	Tests	27
8.10	Tests de performance des algorithmes de correction	28
8.10.1	Principe	28
8.10.2	Fonctionnement	29
8.11	Priorisation des images pour la correction	30
8.11.1	Problématique	30
8.11.2	Principe	31
8.11.3	Pistes de réflexion	31
8.12	Rédaction du rapport de PFE	32
8.13	Préparation de la présentation orale finale	32
9	Analyse critique du projet	33
9.1	Difficultés rencontrées	33
9.2	Bilan par rapport au planning initial	34
10	Conclusion	36
	Glossaire	38
A	Annexes	40
A.1	Cahier de spécifications système	40

Table des figures

5.1	Communication entre le client et le serveur	13
6.1	Gamut CIE RGB	15
6.2	Gamut sRGB	15
6.3	Sensibilité des cônes en fonction de la longueur d'onde	17
8.1	Simulation vision daltonienne	22
8.2	Interface proxy simulation daltonisme	22
8.3	Marquage d'une image	23
8.4	Mise en cache principe page voisines	24
8.5	Mise en cache principe page voisines - fonctionnement	25
8.6	Exemple de planche du test d'Ishihara	27
8.7	Correction image pour le daltonisme - image Ishihara	27
8.8	Correction image pour le daltonisme - image classique	28
8.9	Test de performance - correction LMS	29
8.10	Test de performance - correction LMS et simulation Brettel	30
8.11	Interface proxy priorisation temps de traitement	31
9.1	Diagramme de Gantt prévisionnel	34
9.2	Diagramme de Gantt effectif	34

Introduction

Dans le cadre de notre formation au sein du Département Informatique de l'école Polytechnique de l'Université de Tours, un projet de fin d'études (PFE) est effectué afin de mettre en pratique les connaissances acquises au cours de notre formation.

Ce rapport a pour but de présenter le projet de fin d'études intitulé : « adaptation des images d'un site web pour la compensation du daltonisme ».

Ce projet est réalisé au sein de l'équipe Handicap et Nouvelles Technologies (HaNT) de François Rabelais et est encadré par M. Mohamed Slimane et M. Sébastien Aupetit.

Dans ce rapport, j'introduirai d'abord plusieurs notions essentielles qui concernent ce projet notamment avec la notion d'accessibilité et d'utilisabilité sur le web. Ensuite, afin de comprendre le but du projet, le daltonisme sera présenté. Les notions d'espace colorimétrique ainsi que de protocole HTTP seront aussi abordés afin de mieux comprendre le travail qui a été réalisé. Puis, une fois que les spécifications du système seront faites, je présenterai le travail réalisé lors de ce projet de fin d'études.

L'accessibilité sur le web

2.1 Présentation générale

Dans le domaine du Web, l'accessibilité signifie que les personnes handicapées peuvent utiliser le Web. Cela permet à ces personnes de percevoir, comprendre, naviguer et interagir avec le web ainsi que de pouvoir apporter leur contribution et créer du contenu. L'accessibilité du Web vise toutes les déficiences, qu'elles soient visuelles, auditives, motrices, cognitives, neurologiques ou liées à la parole.

Cependant, de nombreux sites web ne sont pas ou peu accessibles/utilisables pour des personnes souffrants d'un handicap visuel (comme le daltonisme). En effet, des limitations fonctionnelles empêchent des millions de personnes d'utiliser le Web.^[14]

2.2 Importance de l'accessibilité

Dans les secteurs de l'éducation, de l'emploi, du commerce et même du loisir, le web joue un rôle de plus en plus important. C'est pour cela que le Web doit être accessible aux personnes handicapées afin que celles-ci puissent participer activement à la société.

L'accessibilité peut être exigée par la loi. En effet, afin de limiter les problèmes et permettre un accès au web à tous, les associations, lobbys et autres législateurs demandent à ce qu'un certains nombres de principes et normes (WCAG1, WCAG2) soient respectés par les concepteurs de site web. Des lois existent et tentent d'obliger les concepteurs à concevoir des sites web accessibles. Malheureusement, ces lois, normes et principes ne sont pas suffisants.

2.3 Résolution des problèmes d'accessibilité

Rendre un site web accessible à 100% reste utopique pour de nombreuses raisons. Cela serait trop coûteux en temps et argent, trop complexes et trop contraignant. En effet, beaucoup de sites ne sont plus maintenus mais existent toujours ou d'autres sont faits manuellement sans tenir compte de la notion d'accessibilité. Par ailleurs, de nombreux sites web ne sont pas d'origine française, de ce fait, les lois et les contraintes peuvent être différentes voire ne pas exister.

Un site web peut être accessible à une personne ayant l'handicap X sans l'être pour autant avec une autre ayant l'handicap Y. De ce fait, pour répondre au problème d'accessibilité sur des sites web, une solution peut consister à viser une meilleure accessibilité sur ces derniers plutôt que de viser une accessibilité complète.

Pour obtenir une meilleur une meilleure accessibilité (même partielle), il ne faut pas forcément avoir la coopération du concepteur ou du webmaster du site web puisqu'il existe de nombreuses améliorations qui peuvent être effectuées automatiquement (ou semi-automatiquement) avec l'aide d'une tierce personne. Ces améliorations peuvent être réalisées grâce à de l'analyse de documents, de la spécification de pratiques implicites ou de l'étiquetage automatisé par exemple.

2.4 W3C et WAI

Le W3C (Consortium du World Wide Web) a été fondé en 1994 et a pour mission de créer des standards pour le web grâce à des spécifications. L'un des objectifs principaux du W3C est l'accès universel qui signifie, comme son nom l'indique, un accès à tous les usagers afin qu'ils puissent utiliser au mieux internet.

De plus, un des autres objectifs du W3C est l'interopérabilité du web qui signifie que les technologies du Web doivent permettre à tous les matériels informatiques et les logiciels qui accèdent au web de travailler ensemble.^[10]

Pour répondre au problème de l'accessibilité, le W3C a créé la **WAI** (Web Accessibility Initiative) en 1997. La **WAI** propose des solutions techniques pour rendre le Web aux personnes handicapées.

Un certain nombre de recommandations ont été développées par la **WAI**. On peut citer par exemple les Web Content Accessibility Guidelines (**WCAG**) qui permettent de rendre le contenu du web plus accessible ou même les User Agent Accessibility Guidelines (UAAG) qui s'adressent aux développeurs de logiciels qui permettent la navigation et la consultation de contenu.

La notion d'utilisabilité

3.1 Présentation générale

Il faut bien faire la différence entre l'accessibilité d'un site web et l'utilisabilité de ce dernier. En effet, un site web peut très bien être accessible à une personne ayant une déficience visuelle mais cette personne peut être dans l'incapacité d'utiliser ce site.

L'objectif de l'utilisabilité du web est de rendre la navigation sur les sites web plus aisée pour l'utilisateur, sans qu'il ait besoin de suivre un apprentissage spécifique. En effet, les actions de l'utilisateur sur la page web doivent se faire de manière intuitive.

3.2 La notion d'ergonomie

Dans un site web, l'ergonomie consiste à répondre efficacement aux attentes de l'utilisateur afin de leur fournir une navigation plus aisée.

Le problème est que les visiteurs qui visitent le site web n'ont pas tous le même profil. Il faut donc prendre en considération les attentes de l'utilisateur, ses habitudes, son âge, ses équipements ainsi que son niveau de connaissance.

3.3 Règles de Jacob Nielsen

Il existe les dix règles de Jacob Nielsen qui sont l'objet d'un large consensus dans l'évaluation d'un site web.

- **Visibilité de statut du système** : l'utilisateur doit savoir où il est et où il doit aller.
- **Correspondance entre le système et le monde réel** : le système doit parler avec l'utilisateur d'une façon qu'il puisse comprendre facilement.
- **Liberté et contrôle de l'utilisateur** : le système doit offrir à l'utilisateur une liberté par rapport à son espace de travail (résolution d'écran, navigateur...)
- **Homogénéité et cohérence** : les éléments graphiques doivent être traités de façon identiques lorsqu'ils sont utilisés dans un même contexte.
- **Prévention des erreurs** : le système doit prévoir les erreurs que fera l'utilisateur et offrir une instruction appropriée pour éviter que d'autres erreurs se répètent.
- **La reconnaissance** : les directives pour le fonctionnement du système doivent être visibles et facilement récupérables à chaque fois où l'utilisateur en a besoin.

- **Flexibilité et efficacité d'utilisation** : le système devra offrir des options pour les utilisateurs expérimentés et des raccourcis pour les opérations les plus utilisées.
- **Design épuré** : l'interface devrait contenir seulement des informations visuelles pertinentes et mettre l'accent sur le contenu.
- **Fonctions d'aide** : les messages d'erreurs doivent être exprimés dans la langue ordinaire et doivent offrir une solution simple aux utilisateurs.
- **La documentation** : le système doit fournir la documentation nécessaire au besoin.

Ces règles permettent de répondre à la notion d'utilisabilité pour les sites Web.^[7]

Le daltonisme

4.1 Présentation

Le projet concerne les personnes ayant une déficience visuelle et plus particulièrement les personnes atteintes de daltonisme.

Le daltonisme est une maladie répandue dans le monde ; son apparition est due à une anomalie de la rétine. C'est un dysfonctionnement de la vision des couleurs, entraînant le plus souvent la confusion entre le rouge et le vert.

Ce trouble de la vision qu'est le daltonisme est connu depuis le XVIIIème siècle grâce au chimiste anglais John Dalton qui était lui même atteint de cette anomalie.[4]

Le daltonisme est une anomalie dans laquelle un ou plusieurs des trois types de cônes de la rétine oculaire, responsables de la perception des couleurs, sont déficients. Habituellement, le daltonisme est classé comme une infirmité légère. Cependant, il y a des situations où les daltoniens peuvent avoir un avantage sur les individus ayant une vision normale (par exemple pour percevoir certains camouflages basés sur la couleur).

4.2 Causes

Le daltonisme a généralement une origine génétique. Par exemple, lors de la recombinaison génétique (fécondation), il peut y avoir la perte d'un gène ou l'apparition de gènes hybrides.

Cependant, cette anomalie peut parfois survenir suite à une lésion nerveuse, cérébrale ou oculaire.

4.3 Types de d'anomalies

Il existe plusieurs formes de daltonisme partielle, la plus fréquente étant la confusion du vert et du rouge. Les autres formes de daltonisme sont nettement plus rares, comme la confusion du bleu et du jaune, la plus rare de toutes étant la déficience totale de la perception des couleurs, où le sujet ne perçoit que des nuances de gris.

Le daltonisme est classé selon les 3 types de cônes atteints et l'importance du trouble visuel.

4.3.1 Monochromatisme

Ce type d'anomalie correspond à une absence totale de la perception des couleurs. Cette anomalie est très rare et touche une personne sur 40 000. Un individu atteint verra tout ce qui est autour de lui en noir et blanc avec quelques nuances de gris.

4.3.2 Dichromate

Dans ce cas, il y a l'absence du gène, et donc du pigment. Il y a donc une perception de deux couleurs seulement :

- Protanope : perception de vert et du bleu seulement.
- Deutéranope : perception du rouge et du bleu seulement.
- Tritanope : perception du rouge et du vert seulement.

4.3.3 Trichromate anormal

Avec ce type d'anomalie, le gène est hybride donc la sensibilité du pigment est différente. Il y a donc une perception anormale des intensités des trois couleurs :

- Protanomal : lacune au niveau de la perception du rouge.
- Deutéranomal : faible perception du vert.
- Tritanomal : faible perception du bleu.

[6]

Le protocole HTTP

Afin de mieux comprendre le principe de fonctionnement du **proxy**, il semble important d'introduire la notion de protocole HTTP (HyperText Transfer Protocol).

Le but du protocole **HTTP** est de permettre un transfert de fichiers (généralement au format **HTML**) localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur web .

La communication entre le client et le serveur se fait de cette manière :

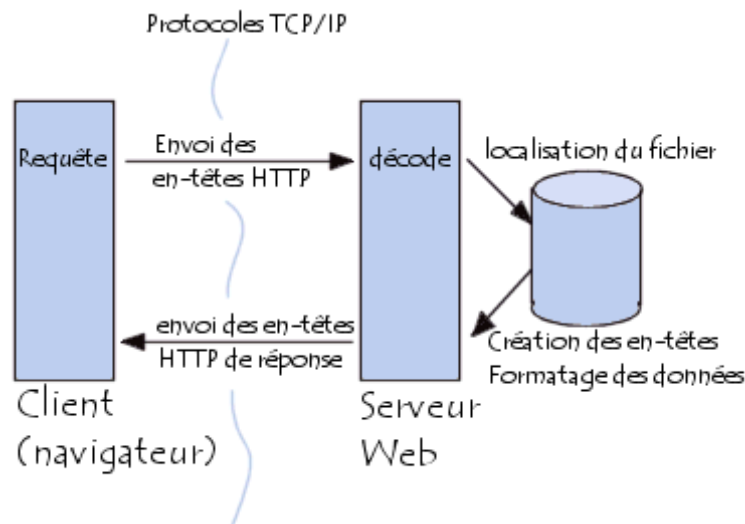


FIGURE 5.1 – Communication entre le client et le serveur

On peut voir que cette communication se fait en deux temps :

- Le navigateur effectue une requête HTTP
- Le serveur traite la requête puis envoie une réponse HTTP

Une requête **HTTP** est un ensemble de lignes envoyé au serveur par le navigateur et qui comprend :

- Une ligne de requête : précise le type de document demandé, la méthode qui doit être appliquée, et la version du protocole utilisée
- Les champs d'en-tête de la requête : lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client
- Le corps de la requête : ensemble de lignes optionnelles permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire[2]

Dans ce rapport, l'introduction du protocole **HTTP** va permettre de mieux comprendre le principe du **proxy** SWAP notamment avec la communication entre le navigateur web, internet et le **proxy**.

La notion d'espace colorimétrique

Nous allons introduire la notion d'espace colorimétrique. Cela est justifié car une grande partie du projet a été consacrée à la manipulation d'images. Il est donc très important de comprendre comment les couleurs sont représentées. En effet, pour simuler la vision daltonienne et corriger les pertes visuelles d'un daltonien, il faudra par exemple changer le système de représentation des couleurs.

6.1 Définition

Un espace colorimétrique ou espace de couleur associe des nombres aux couleurs visibles. Ces nombres se présentent généralement sous la forme de triplets. Chaque couleur de lumière peut donc être caractérisée par un point dans un espace à trois dimensions. Par exemple, l'Adobe RGB et le sRGB sont 2 espaces colorimétriques différents, mais sont basés sur le même modèle colorimétrique (RVB). [15]

6.2 Le RGB

Le RGB(Red, Green, Blue), que l'on nomme aussi en français RVB (Rouge Vert Bleu) est un système colorimétrique informatique permettant de représenter n'importe quelle couleur sous la forme d'une composante de ces trois couleurs.

Les trois couleurs (Rouge Vert Bleu) sont trois couleurs primaires. L'œil humain contient trois types de cônes. Ces trois couleurs correspondent aux trois longueurs d'ondes répondant à ces trois types de cônes. De plus, si l'on additionne ces trois couleurs, cela donnera du blanc pour l'œil humain.

Il existe une infinité de systèmes RVB. Ces derniers dépendent du choix de la teinte et de la saturation de chaque composante. Un diagramme de chrominance (x,y) est utilisé. Dans ce diagramme, les trois points correspondants définissent un gamut triangulaire. A partir des trois couleurs primaires on ne peut reconstituer seulement les couleurs qui sont situées dans ce triangle.

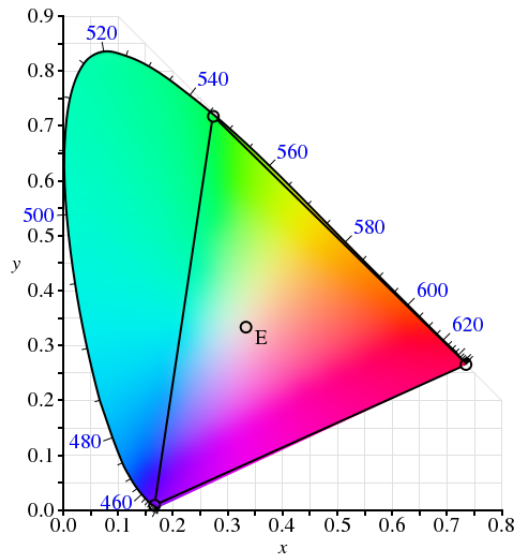


FIGURE 6.1 – Gamut CIE RGB

On peut donc voir qu'un système RVB ne peut décrire toutes les couleurs visibles par l'œil humain car on remarque que le **gamut** triangulaire est contenu dans l'ensemble des couleurs visibles. Le système RVB ne couvre donc pas toutes les couleurs.[16]

6.3 Le sRGB

Le sRGB, que l'on nomme aussi standard RGB, est un espace de couleur rouge vert bleu créé par Hewlett-Packard et Microsoft. Il représente un standard officiel des images numériques.

Le RVB est ce qu'on nomme un modèle colorimétrique qui est une famille d'espaces colorimétriques fondée sur le même principe. Le sRGB est un espace colorimétrique bien défini. Son but est d'utiliser un profil RVB unique pour tous les systèmes ou appareillages basés sur le RVB.

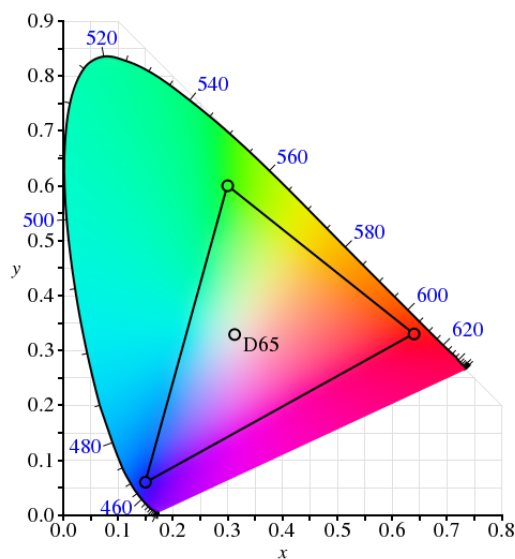


FIGURE 6.2 – Gamut sRGB

Le triangle sRGB semble contenir peu de couleurs par rapport à l'ensemble des couleurs visibles. De plus, plus les couleurs se trouvent proches des limites du visible, plus elles sont rares.

Le sRGB est devenu le standard pour la publication d'image sur internet.[3]

6.4 CIE Lab

Chaque individu peut percevoir différemment les couleurs. Ces dernières peuvent aussi être affichées différemment selon les périphériques d'affichage.

Des standards ont été défini par la Commission Internationale de l'Eclairage (CIE). Ces standards permettent de définir une couleur indépendamment des périphériques utilisés.

La CIE a élaboré le système colorimétrique xyY. Ce système représente les couleurs selon leur chromaticité (axes x et y) et leur luminance (axe Y). Ce modèle de représentation, CIE Lab, est aussi plus précisément nommé $L^*a^*b^*$.

La signification de $L^*a^*b^*$ est la suivante :

- La composante L^* est la luminance (ou la clarté), qui va de 0 (noir) à 100 (blanc).
- La composante a^* représente une gamme couleur allant du vert au rouge et passant par le gris.
- La composante b^* représente une gamme couleur allant du bleu au jaune et passant par le gris.

L'intégralité du spectre visible par l'œil humain est donc couvert par le mode Lab. Ce mode permet donc de décrire l'ensemble des couleurs visibles. [1]

6.5 LMS

Il s'agit d'un espace de couleur qui représente la réponse des trois types de cônes de l'œil humain.

Il existe trois types de cônes différents. Ces trois types de cônes correspondent à des plages de longueur d'onde différentes : les cônes L(long, autour du rouge), les cônes M(medium, autour du vert) et les cônes S(short, autour du bleu). Associés avec les bâtonnets qui sont sensibles à la luminosité, ces trois types de cônes nous permettent de voir l'ensemble des couleurs.

Ces courbes montrent la sensibilité des trois types de cônes en fonction de la longueur d'onde.

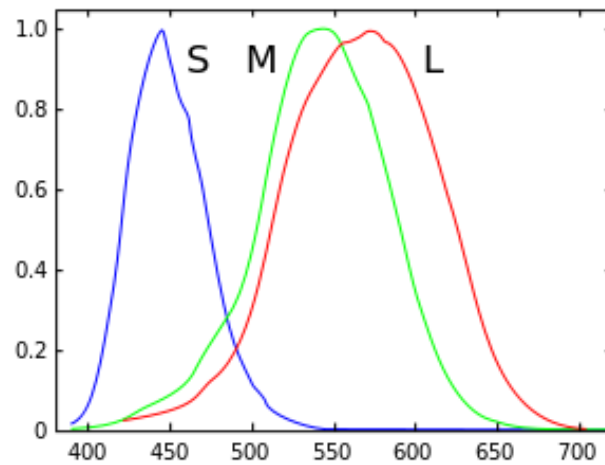


FIGURE 6.3 – Sensibilité des cônes en fonction de la longueur d'onde

Par ailleurs, le passage des valeurs RGB d'un pixel dans l'espace de couleur LMS est relativement simple car il ne s'agit que d'une multiplication entre chaque valeur RGB et une matrice carrée de trois lignes et trois colonnes.[\[17\]](#)

Spécifications système

7.1 Contexte

Au départ, le web avait été mis en place pour pouvoir être utilisé par tout le monde. Mais il s'avère qu'aujourd'hui, cet objectif n'est pas atteint. Actuellement, de nombreux sites web ne sont pas ou peu accessibles/utilisables pour des personnes souffrants d'un handicap visuel (comme le daltonisme). Ainsi, des limitations fonctionnelles empêchent des millions de personnes d'utiliser le Web. Ceci peut être expliqué lorsque l'on sait que le développement d'un site web est soumis à un ensemble de contraintes (financières ou d'ergonomies). De plus, il n'est pas rare qu'un site web soit délaissé par celui qui a conçu ce site.^[11]

Ici, le but est de rendre plus simple la navigation de l'utilisateur sur les sites web sans qu'il ait besoin de suivre un apprentissage spécifique. Les actions de l'utilisateur sur la page web doivent se faire de manière intuitive.

L'équipe HaNT de l'Université François Rabelais tente de mettre en place un outil facilitant l'accès au web pour les personnes handicapées. L'objectif de cet outil est de traiter les pages web afin de les rendre plus accessibles.

Mon projet de fin d'études est donc directement en relation avec l'outil mis en place par l'équipe HaNT puisque le sujet de mon projet est d'adapter les images d'un site web pour la compenser les pertes visuelles dues au daltonisme.

7.2 Objectifs

L'objectif est de permettre l'utilisation d'un web non accessible/non utilisable à des personnes ayant une déficience visuelle.

Dans ce projet, on cherche à compenser les pertes visuelles dues au daltonisme sur les images. L'ensemble du projet sera réalisé en Java au sein de la plateforme SWAP (Smart Web Accessibility Proxy).

Pour connaître l'ensemble des spécifications système de ce projet, voir en annexe(A).

Travail réalisé

8.1 Recherches et documentation

Tout d'abord, le projet a commencé par une longue phase de recherche et de documentation. Cela a consisté à se documenter sur tout ce qui tourne autour du daltonisme afin de comprendre comment fonctionne cette déficience visuelle. Il a fallu étudier la notion d'accessibilité et d'utilisabilité sur le web, puis trouver comment modifier les couleurs d'une image afin de passer de la vue d'une « personne normale » à une personne daltonienne. Il fallait aussi comprendre comment une couleur est représentée au niveau du codage (RGB par exemple), mais aussi découvrir les différents algorithmes existants permettant de simuler la vue d'une personne daltonienne.

Une synthèse résumant toutes les recherches a été faite. Cette synthèse a servi à la rédaction du rapport final de ce projet. Son intérêt était également de réaliser une bibliographie avec tous les liens et documents qui avaient été utilisés pour ces recherches. L'intérêt de cette bibliographie était de pouvoir retrouver facilement les différents liens qui m'avaient permis de comprendre mon projet de fin d'études.

8.2 Installation de l'environnement de développement

Cette tâche a consisté à installer tous les outils nécessaires au développement sur la plateforme SWAP. Différentes installations ont dû être effectuées comme l'installation de l'**IDE** Eclipse ainsi qu'un logiciel de gestion de version comme SVN. De plus, plusieurs plugins ont été installés avec Eclipse comme Subversive et **Maven** 3.

Par ailleurs, il a fallu récupérer le projet SWAP sur le serveur mis à disposition par nos encadrants. Pour cela fallait utiliser le logiciel de gestion de versions SVN.

8.3 Analyse de l'existant de la plateforme SWAP

Prendre en main le code source du projet SWAP était une étape importante du projet. Ce dernier étant composé de multiples classes, il faut pouvoir comprendre comment est structuré ce projet ainsi que savoir comment fonctionnent toutes ces classes.

Une synthèse expliquant le fonctionnement général de la plateforme SWAP a été rédigée. Cette synthèse indique comment les différents composants de la plateforme SWAP interagissent entre eux et le principe de chacun de ces composants.

8.4 Rédaction du cahier de spécifications

Dans ce projet, un cahier de spécifications devait être rédigé. Ce document permet de décrire le projet d'un point de vue général d'abord, puis de spécifier l'architecture du système ainsi que les différentes fonctionnalités. Enfin, ce document permet de découper le projet en tâches.

Le livrable est un document de type PDF réalisé avec le système de composition de documents nommé Latex.

Dans ce cahier de spécifications, un planning prévisionnel avait été mis en place. Ce planning a permis de comparer ce qui a été réellement fait par rapport à ce qui avait été prévu.

8.5 Préparation de la présentation orale à mi-parcours

Une présentation orale a été faite le 5 Février 2013 avec la présence des encadrants et des autres élèves travaillant au sein de l'équipe Handicap et Nouvelles Technologies.

Cette présentation orale permettait de faire un bilan de mi-parcours du projet. Cet oral a été utile puisqu'il m'a permis de présenter ce que j'avais compris du projet, de me confirmer que je ne faisais pas fausse route et enfin de me préparer en vue de la soutenance finale.

Il a donc fallu préparer cette présentation orale. De plus, un poster permettant d'expliquer brièvement le projet a été réalisé. Ce poster permettait aux visiteurs de l'école, lors des journées portes ouvertes, d'être informé sur quoi travaille l'équipe HaNT actuellement.

8.6 Simuler une vue daltonienne

Avant de commencer à étudier le principe de la compensation des pertes visuelles dues au daltonisme, ma première grande tâche de développement était de simuler la vue d'un daltonien sur le web. Le fait de passer par cette étape intermédiaire avant de traiter le sujet de la correction du daltonisme est justifiée car cela m'a permis de prendre en main le principe de manipulation des images.

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de simuler une vue daltonienne.

8.6.1 Outils de simulation

D'abord, pour comprendre le principe de la simulation daltonienne, j'ai cherché s'il existait des outils qui permettait de faire cette simulation. J'ai alors découvert différents outils existants :

- **Vischeck** est un site web qui permet d'uploader une image et de la visualiser comme un daltonien pourrait le faire. [13]
- **Color Oracle** est un logiciel à télécharger qui, une fois lancé, permet d'avoir une vision daltonienne sur son écran (ici nous ne sommes pas limité à la vision d'une page web ou d'une unique image). [12]
- **Chrome Daltonize** est une extension disponible avec le navigateur web Google Chrome. La simulation doit être refaite dès qu'une nouvelle page est chargée par le navigateur. [5]

Pour chacun de ces outils, il est possible de choisir le type de daltonisme que l'on souhaite simuler : la vision protanope, deutéranope et tritanope. De plus, certains de ces outils comme Vischeck et Chrome Daltonize sont aussi capable de corriger les images.

8.6.2 Manipulation des images

Tout d'abord, il est important de savoir que lorsqu'une page web est traitée par le proxy, un système est mis en place (un handler) afin d'intercepter chaque image présente sur la page. Pour chaque image, le proxy va récupérer un flux correspondant à l'image qui vient d'être interceptée. Au départ, pour manipuler une image, j'avais donc en entrée un flux correspondant à celle-ci.

La question qui se posait était alors de savoir comment manipuler cette image. Je me suis donc renseigné sur la bibliothèque **OpenCV**, mais j'ai finalement opté pour l'utilisation de **ImageIO** qui est une interface de programmation(**API**) de Java. Cette solution semble être la plus appropriée.

Pour manipuler l'image, il suffit de lire le flux grâce à une fonction Java existante puis de parcourir l'ensemble des pixels qui composent l'image. En effet, pour simuler la vision daltonienne d'une image, il faut modifier tous les pixels un par un.

8.6.3 Principe de la simulation

La simulation de la vue daltonienne se fait donc comme ceci :

- on récupère un flux correspondant à l'image
- on lit le flux grâce à une fonction Java
- on crée une nouvelle image vierge
- on parcourt l'ensemble des pixels de l'image
- pour chaque pixel on applique une transformation pour passer à la vision daltonienne
- on récupère la valeur du pixel transformé
- on reconstruit la nouvelle image à partir de tous les pixels transformés
- on envoie la nouvelle image au navigateur

Cependant, tout ceci n'est pas aussi simple qu'il n'y paraît. En effet, chaque image possède son propre espace de couleur. Généralement, l'encodage le plus fréquent est le RGB mais parfois certaines images peuvent être codées différemment notamment avec la présence de la transparence.

8.6.4 Fonctionnement de la simulation

Sur la plateforme SWAP, une fonction est déjà présente et permet de simuler la vision daltonienne d'un pixel avec l'algorithme de Kuhn.

De plus, afin de pouvoir appliquer la simulation, il ne faut pas oublier de changer l'espace de couleur si besoin. Par exemple, pour l'algorithme de Kuhn implémenté sur la plateforme SWAP, il faut que l'encodage du pixel soit en CIE Lab. Évidemment, une fois la simulation faite, on repasse le pixel dans l'espace RGB. Des fonctions permettant de changer d'espace de couleur comme entre le RGB et le CIE Lab sont déjà implémentées sur la plateforme.

La simulation fonctionne très bien avec l'algorithme déjà en place sur la plateforme SWAP, mais je me suis intéressé à un autre algorithme de simulation qui lui n'était pas déjà implémenté. Après avoir recherché et étudié l'algorithme de Brettel, j'ai pensé qu'il pourrait être intéressant d'implémenter moi-même cette méthode. Au final, mon implémentation donne des résultats identiques à ceux de la simulation précédemment mise en place.[9]

Voici un exemple de la transformation d'une image normale en une image simulée avec la vue daltonienne.



FIGURE 8.1 – Simulation vision daltonienne

8.6.5 Interface sur la plateforme SWAP

Une interface a été ajoutée sur la plateforme SWAP. Cette interface permet à l'utilisateur du proxy de choisir le type de simulation qu'il souhaite.

Settings

> Root > Welcome > Configuration

Available configuration modules

Proxy	Configure the proxy component
Cache management	Configure the cache manager
Daltonism simulation management	Choose de type of daltonism

Management of the type of daltonism (simulation)

Choose the type of daltonism :

Deuteranope

Save changes



FIGURE 8.2 – Interface proxy simulation daltonisme

Cela permet de ne pas à avoir changer manuellement dans le code source le type de simulation du daltonisme.

8.6.6 Tests

Des tests ont été effectués au fur et à mesure de l'avancement du développement. Ces tests consistaient à naviguer aléatoirement sur différents types de sites web pour vérifier que la simulation daltonienne était effective sur toutes les images.

8.6.7 Intérêt

Grâce à cette étude de simulation de la vue daltonienne, la mise en place de la correction a été beaucoup plus rapide. En effet, le principe de manipulation des images est identique pour la récupération de l'image, le parcours de chaque pixel ou même pour l'envoi de l'image transformée au navigateur. De plus, on verra par la suite que pour certains algorithmes de correction des images, il faut passer par la simulation de la vue daltonienne.

8.7 Marquage des images

Le marquage des images est une étape qui a été faite en prévision de la priorisation des images lors de la correction permettant de compenser les pertes visuelles dues au daltonisme. 8.11

8.7.1 Fonctionnement

Un marquage des images a été effectué afin de pouvoir identifier chaque image appartenant à une page. Ce marquage consiste à ajouter à la fin de l'url de l'image un identifiant unique généré pour chaque page grâce à la classe Java UUID(Universal Unique Identifier).

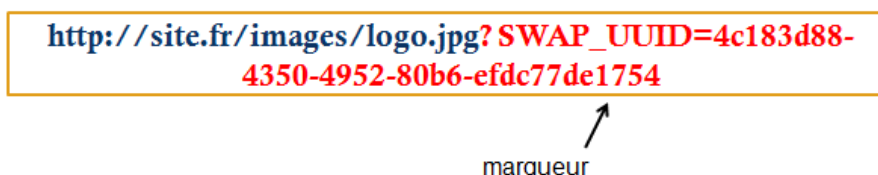


FIGURE 8.3 – Marquage d'une image

Pour chaque page un nouvelle identifiant unique est généré. Cet identifiant est aussi ajouté dans les cookies du navigateur. Cela permet de savoir à quelle page appartient une image donnée. Ceci sera utile plus tard lorsque l'on procédera à la mise en cache des pages et des images voisines de la page HTML courante. 8.8

8.7.2 Intérêt

L'intérêt d'un tel marquage est qu'avec ce système, il est désormais possible de marquer l'image que l'on souhaite. Cela pourra être très utile pour l'étape finale du projet qui consiste à prioriser les images pour la correction du daltonisme.

8.8 Mise en cache des images

Une des grandes étapes du projet au niveau du développement a été la mise en cache des images des pages voisines.

8.8.1 Principe

Les pages voisines correspondent aux pages qui sont accessibles depuis la page qui est en cours de visualisation dans le navigateur internet.

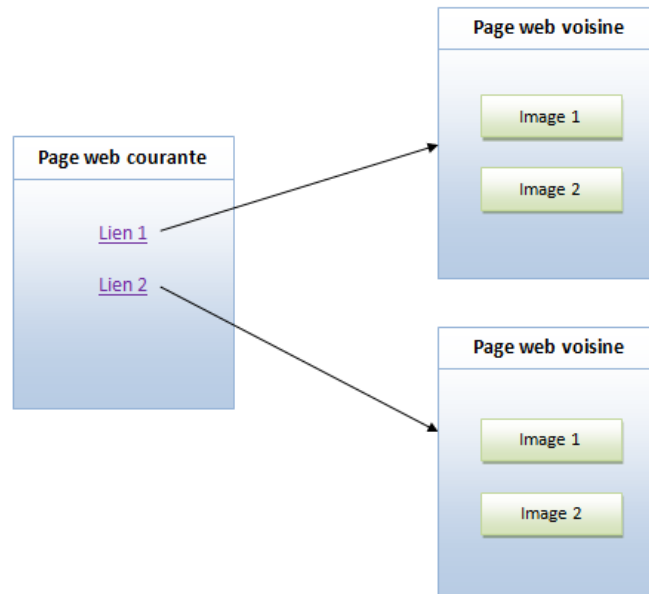


FIGURE 8.4 – Mise en cache principe page voisines

8.8.2 Fonctionnement

Le fonctionnement de la mise en cache se fait de cette façon :

- on parse la page HTML qui a été demandée par l'utilisateur
- on récupère tous les liens de cette page (liens http ://)
- on met en cache toutes les pages voisines grâce à ces liens
- pour chacun de ces liens on crée un **thread** qui va récupérer toutes les images de la page
- pour chacune des images on crée un thread qui va mettre en cache l'image

Cette mise en cache peut-être schématisée comme ceci :

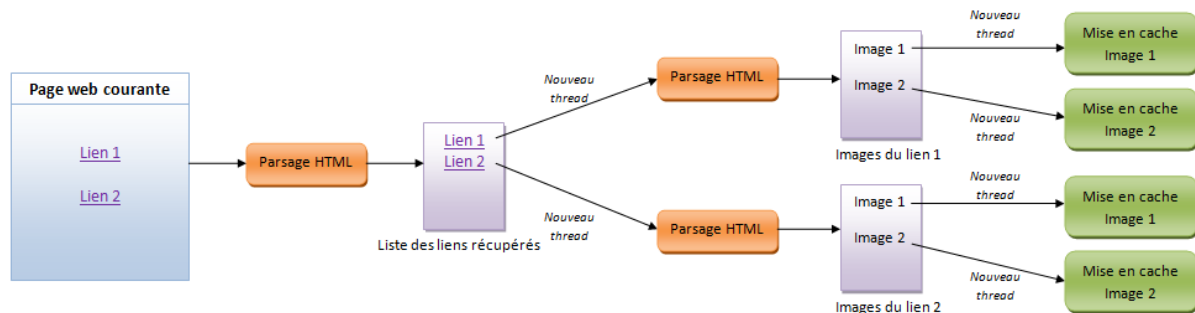


FIGURE 8.5 – Mise en cache principe page voisines - fonctionnement

L'utilisation des threads permet de pouvoir exécuter plusieurs tâches en parallèle. De plus, afin de ne pas avoir trop de traitements en parallèle, j'ai utilisé l'interface Executors de Java qui permet de mettre en place un pool(groupement) de threads. En accord avec Mr Aupetit, j'ai fixé de pool de 4 threads. Avec ceci, il n'y aura au maximum que 4 threads qui s'exécuteront en parallèle. Si un nouveau thread est ajouté alors qu'il y a déjà 4 threads en cours d'exécution, une file d'attente va se mettre en place, et lorsqu'un des threads aura terminé son exécution, il laissera la place à un des threads présents dans la file d'exécution.

Enfin, j'ai indiqué que la priorité des threads permettant la mise en cache des images seraient inférieure à celle des autres threads du proxy.

8.8.3 Intérêt

Cette mise en cache permet au proxy de ne pas avoir à recharger les images lorsqu'il souhaite les afficher et donc de gagner en temps de chargement.

8.8.4 Tests

Pour tester cette mise en cache, il suffisait de naviguer sur différentes pages web choisies aléatoirement. Afin de vérifier que la mise en cache était bien effectuée, la plateforme SWAP possède une interface permettant de visualiser le contenu du cache.

8.9 Correction des images

Une des étapes importantes du projet était de corriger les images afin que les troubles visuels des daltoniens soient compensés.

Cette correction d'image suit directement l'étape de la simulation de la vue daltonienne, d'abord car le principe de manipulation de l'image reste le même (récupération du flux, lecture...), et ensuite car la mise en place de certains algorithmes de correction obligent à passer par la simulation daltonienne de l'image.

Aucune fonction permettant de corriger une image n'existait déjà sur la plateforme SWAP, contrairement à la fonction de simulation daltonienne d'une image. De ce fait, de nombreuses recherches ont été effectuées afin de comprendre comment il était possible de corriger une image. Ces recherches ont consisté à trouver des algorithmes permettant d'effectuer cette correction puis à comprendre ces algorithmes.

8.9.1 Algorithmes de correction

Différents algorithmes de correction existent.

8.9.2 Passage par l'espace de couleur LMS

L'algorithme de correction des images qui utilise le passage des valeurs RGB dans l'espace de couleur LMS (6.5) se décompose en 5 étapes. Pour chaque pixel RGB de l'image, on va :

- convertir le pixel RGB dans l'espace de couleur LMS
- simuler la vision daltonienne avec les valeurs LMS
- convertir les valeurs LMS simulées en RGB
- calculer l'erreur entre les valeurs RGB initiales du pixel et les valeurs RGB du pixel simulé précédemment (il s'agit de faire une soustraction entre le RGB initial et le RGB simulé)
- modifier les erreurs des trois valeurs RGB du pixel

Finalement, les valeurs RGB du pixel correspondront aux valeurs RGB initiales additionnées aux valeurs RGB modifiées.

Avant de me lancer dans l'implémentation de ce type d'algorithme de correction, Mr Aupetit m'avait indiqué que ce type de correction, avec le passage des valeurs RGB dans l'espace de couleur LMS, allait être difficile à mettre en place car les différentes matrices utilisées dans cette correction sont rarement les mêmes. Une fois que l'algorithme était mis en place, j'ai donc testé de nombreuses matrices différentes que je pouvais trouver sur le web.[18]

Au final, après une longue période de tests, je pense avoir trouvé des matrices qui me permettent d'avoir un bon résultat.

Par ailleurs, comme expliqué précédemment (8.6.4), deux algorithmes de simulation de la vue daltonienne existent : la simulation de Kuhn déjà implémentée sur le proxy par Mr Aupetit et la simulation de Brettel que j'ai implémenté moi-même. Il faut savoir que la correction de l'image est identique avec les deux algorithmes.

8.9.3 Passage par l'espace de couleur CIE Lab

Un autre algorithme de correction d'image existe. Celui-ci a un fonctionnement complètement différent de l'algorithme présenté précédemment.

Pour cette correction il est nécessaire de passer les valeurs de chaque pixel RGB dans l'espace de couleur CIE Lab.(6.4) Pour faire ceci, une fonction est déjà implémentée sur la plateforme SWAP.

Cet algorithme se décompose comme ceci :

- quantification de l'image
- optimisation "Mass Spring" des couleurs quantifiées
- reconstruction des couleurs finale depuis les couleurs optimisées

La quantification consiste à réduire le nombre de couleurs présentes sur l'image.[8]

Cet algorithme de correction n'a pas été implémenté entièrement. En effet, seule la première étape consistant à quantifier l'image a été effectuée.

8.9.4 Tests

Afin de pouvoir tester les différents algorithmes de correction des images on pourrait bien sûr prendre des images aléatoirement. Cependant, pour réellement voir si la correction de l'image fonctionne correctement, il a semblé justifié d'utiliser les planches du test d'Ishihara. Ces planches sont utilisées pour la détection des déficiences visuelles. Ce test est donc fréquemment utilisé pour déceler le fait qu'un individu soit daltonien ou non.

Voici un exemple de planche du test d'Ishihara :

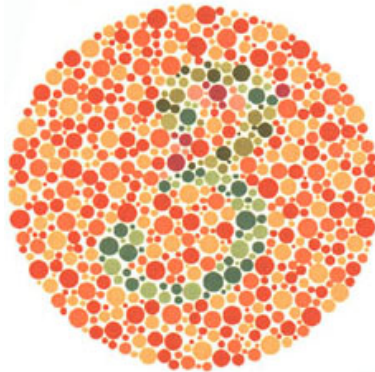


FIGURE 8.6 – Exemple de planche du test d'Ishihara

Sur cette planche, les individus ayant une vision normale doivent clairement percevoir le chiffre 3, alors que les individus ayant une déficience du rouge et du vert percevront le chiffre 5.

L'intérêt d'utiliser ces planches du test d'Ishihara semble évident. Si on applique un algorithme de correction sur une planche, alors un individu daltonien devrait pouvoir percevoir le même chiffre qu'un individu normal.

Si on applique l'algorithme de correction avec le passage des valeurs RGB dans l'espace de couleur LMS (6.5) sur une planche du test d'Ishihara, on va obtenir ceci :

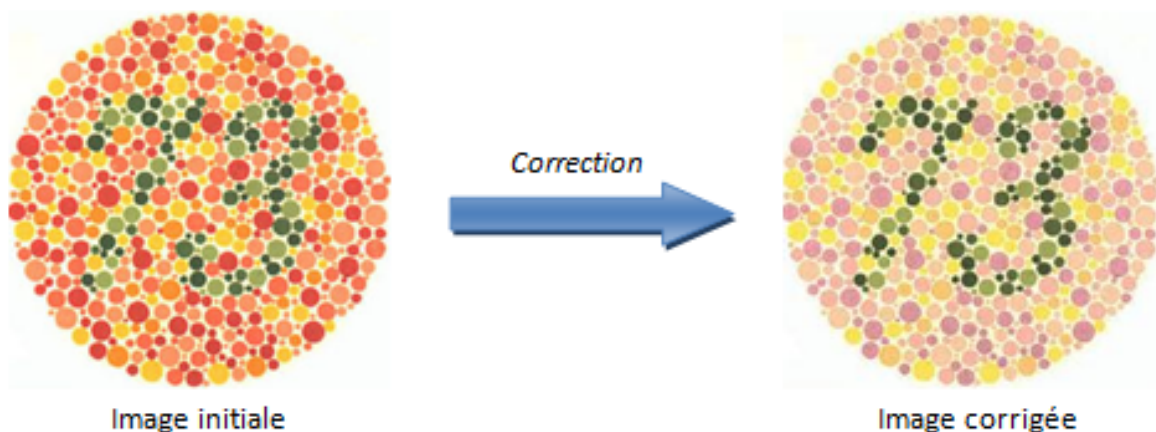


FIGURE 8.7 – Correction image pour le daltonisme - image Ishihara

Sur l'image de gauche, un individu ayant une vision normale doit percevoir le nombre 73 alors qu'un individu daltonien ne pourra rien percevoir ou alors de façon erronée. Lorsque l'on applique l'algorithme

de correction, un individu daltonien pourra percevoir le nombre 73. Cela montre donc que l'algorithme fonctionne donc bien.

Pour une image classique, voici ce que donne l'algorithme de correction pour un individu deutéranope :

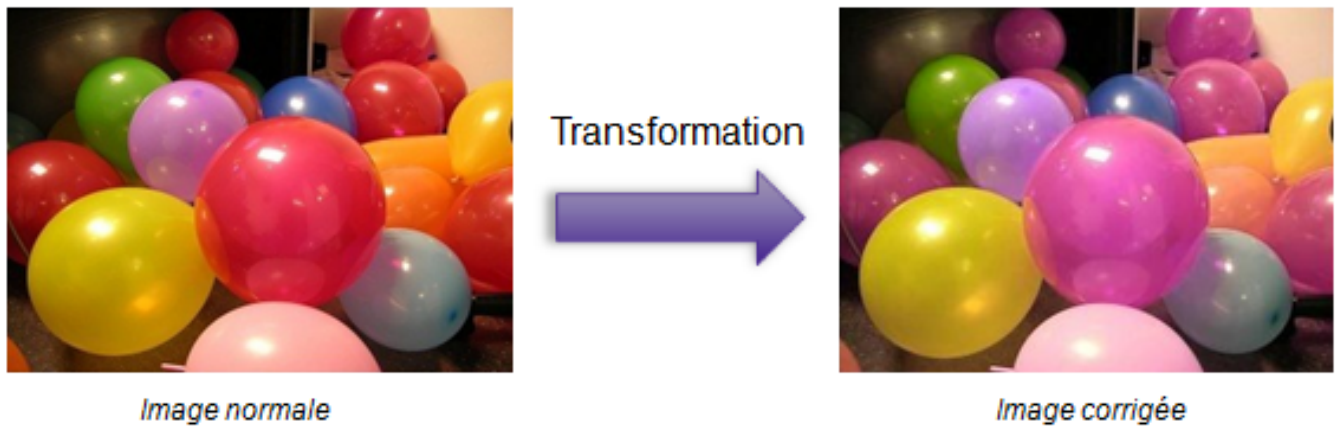


FIGURE 8.8 – Correction image pour le daltonisme - image classique

Étant moi-même daltonien, il était donc aisé de pouvoir tester l'algorithme de correction mis en place. Pour l'exemple précédent avec la planche du test d'Ishihara, j'ai beaucoup de difficulté à discerner clairement un nombre sur l'image initiale. Une fois la correction effectuée sur cette image, j'arrive aisément à distinguer le nombre 73 présent sur la planche.

De plus, afin de vérifier que l'algorithme mis en place fonctionne correctement, je pouvais aussi comparer mes résultats avec les outils existant permettant de corriger les images pour les daltoniens comme Vischeck ou avec l'extension du navigateur Google Chrome : Chrome Daltonize. 8.6.1

8.10 Tests de performance des algorithmes de correction

Après avoir implémenté différents algorithmes de correction des images pour compenser les pertes visuelles dues au daltonisme, la question qui se posait était alors de savoir quel algorithme était le plus performant.

8.10.1 Principe

Pour savoir quel algorithme est le plus performant, le principe était de créer une page **HTML** contenant des images aléatoires de tailles différentes. Pour faire ceci, j'ai créé un script bash qui me générerait 10 images aléatoirement, toutes différentes, et toutes de taille différentes. Pour générer aléatoirement des images, j'ai utilisé ImageMagick. Ce dernier est un logiciel libre qui comprend une bibliothèque et un ensemble d'utilitaires en ligne de commande.

8.10.2 Fonctionnement

Le fonctionnement de ces tests a consisté à lancer un algorithme de correction sur la page **HTML** créée grâce au script bash.

Ces tests fonctionnent de la façon suivante : on crée un fichier avec l'extension `.dat` que l'on remplit dès qu'une image est corrigée. On y insère alors la taille de l'image et le temps CPU (Central Processing Unit) utilisé pour corriger cette même image. Une fois que toutes les images sont corrigées, on dessine un graphique avec toutes les données contenues dans le fichier temporaire précédemment créé.

Le graphique créé permet de pouvoir observer les temps de traitement pour les images de la page HTML générée grâce au script bash. Ce graphique aura sur l'axe des abscisses les différentes tailles des images générées aléatoirement, et sur l'axe des ordonnées le temps de traitement. La taille d'une image est calculée simplement en multipliant la longueur et la largeur, cela correspond donc au nombre total de pixels de l'image. Le temps de traitement est calculé grâce au temps CPU plutôt que du temps d'horloge.

Lors de la réalisation des premiers tests je pouvais remarquer que le temps de traitement pour la première image (la plus petite) était largement supérieur à toutes les autres. Ceci était incohérent car il n'est pas normal que le temps de correction de la plus petite image soit supérieur aux autres images plus grandes. Ceci s'explique car pour la première image, le code source permettant la correction va être d'abord compilé, alors que pour les images suivantes le code source sera interprété par la machine virtuelle Java (JVM). Cette première phase de compilation est appelée "warm up" et nécessite donc plus de temps de traitement que lorsque le code est déjà compilé.

Pour résoudre ce problème il suffit d'exécuter la correction de toutes les images au moins une fois sans tenir compte des temps de traitement.

Les tests ont été répétés 50 fois et la page HTML contenait 10 images de tailles différentes. Voici le résultat de ces tests avec le premier algorithme de correction mis en place qui utilise le passage des valeurs RGB dans l'espace de couleur LMS.

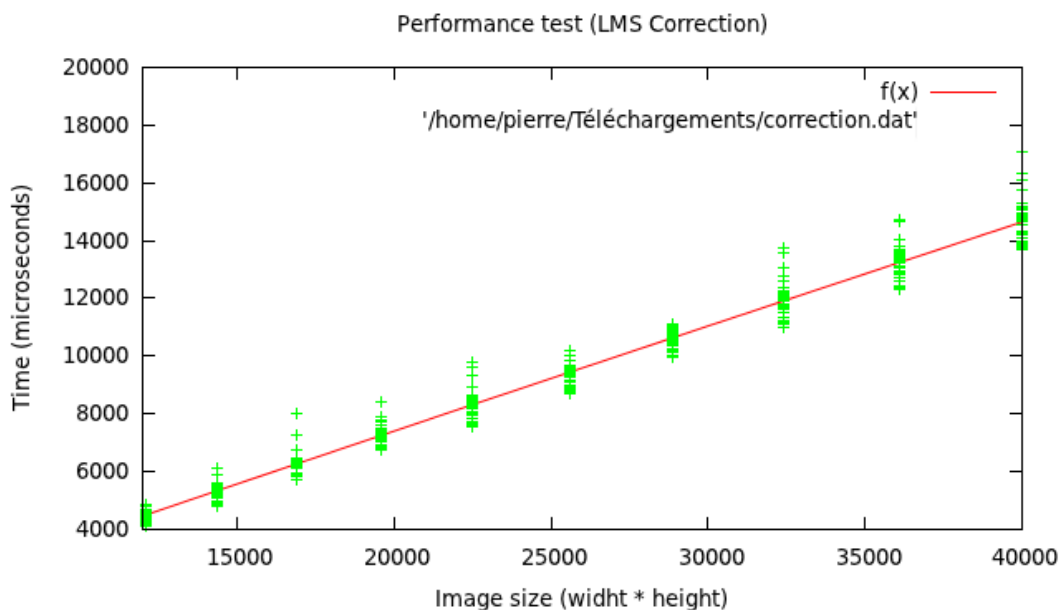


FIGURE 8.9 – Test de performance - correction LMS

Ensuite, le résultat des tests avec le deuxième algorithme mis en place qui utilise toujours le passage des valeurs RGB dans l'espace de couleur LMS mais en utilisant la simulation de Brettel.

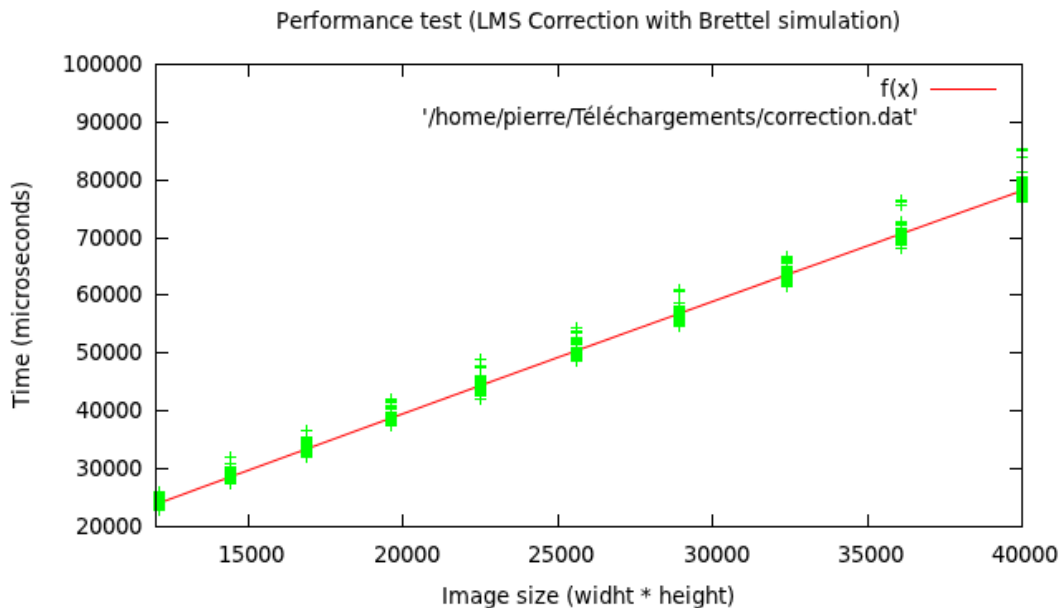


FIGURE 8.10 – Test de performance - correction LMS et simulation Brettel

On remarque que le temps de traitement est largement supérieur pour la correction d'image utilisant la simulation de la vue daltonienne de Brettel. En effet, pour une image d'environ 20000 pixels, la correction prend 40000 microsecondes si on utilise la simulation daltonienne de Brettel alors qu'avec le premier algorithme le temps de traitement est réduit à environ 7000 microsecondes. Cet écart indique clairement qu'il vaut mieux laisser de côté la correction utilisant la simulation de Brettel.

8.11 Priorisation des images pour la correction

8.11.1 Problématique

Après avoir mis en place des algorithmes permettant de compenser les pertes visuelles dues au daltonisme et après avoir comparé les temps de traitement de chacun des différents algorithmes, la question de priorisation des images se pose. En effet, si une page web contient beaucoup d'images, le temps de traitement de toutes ces images sera long et l'affichage de la page sur le navigateur sera évidemment beaucoup plus long aussi. Cela peut-être un réel problème pour l'utilisateur du proxy.

8.11.2 Principe

Le principe de la priorisation des images pour la correction est de savoir quelle image le proxy va pouvoir corriger ou non.

Afin que l'utilisateur puisse gérer lui même le temps de traitement maximal des images du proxy, une interface a été créée.

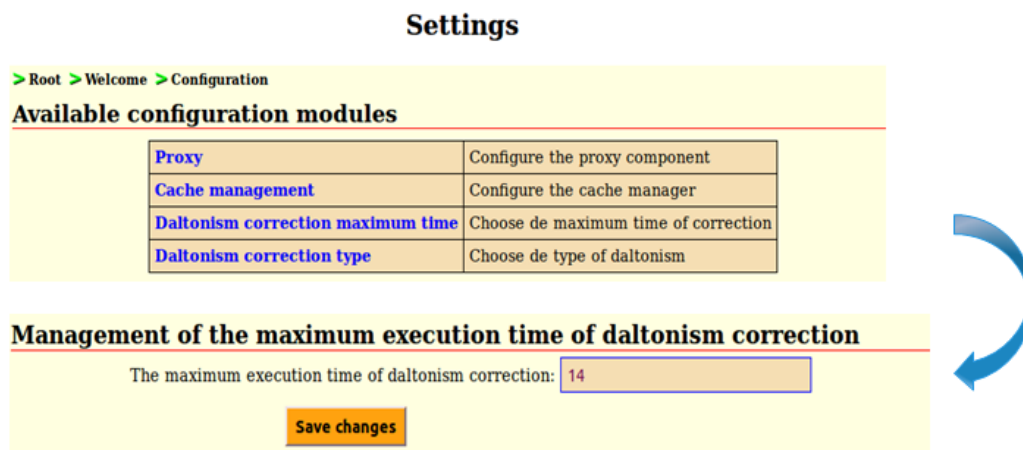


FIGURE 8.11 – Interface proxy priorisation temps de traitement

Grâce à cette interface l'utilisateur peut choisir le nombre de secondes que prendra le proxy pour corriger les images.

8.11.3 Pistes de réflexion

La mise en place de la priorisation pour la correction des images n'a pas pu être faite par manque de temps. Cependant j'ai pu réfléchir à la façon dont cela pouvait être fait.

Le fonctionnement basique reviendrait à corriger les images jusqu'à ce que le temps de traitement maximal défini par le proxy soit atteint. Evidemment, si ce temps est atteint alors que le proxy est en train de corriger une image, il faudra finir la correction de celle-ci.

Cependant, cette approche ne répond pas à la problématique suivante : "Quelles sont les images les plus importantes de la page?". En effet, pour que la priorisation soit la meilleure possible, il faudrait pouvoir déterminer qu'une image est plus importante qu'une autre.

Pour répondre à cette problématique, différentes pistes peuvent être abordées. Ces pistes sont énoncées par ordre de pertinence, c'est à dire que la première piste citée est la plus pertinente.

La taille de l'image

On pourrait prendre en compte la taille de l'image. En effet, plus une image est de grande taille plus elle sera visible par l'utilisateur et elle peut donc être considérée comme étant plus importante que les autres images plus petites de la page.

La position de l'image

Chaque image présente dans une page HTML peut être décrite par une balise "ALIGN" qui indique sa position. On pourrait étudier la position de l'image dans la page afin de déterminer son importance. Une image située en haut et au centre de la page pourrait sembler plus importante qu'une image située en bas à droite de la page.

La description de l'image

Chaque image présente dans une page HTML peut être décrite par une balise "TITLE" ou "ALT" qui permet d'afficher une description de l'image. Ceci pourrait aussi être pris en compte et on peut se demander si une plus grande description d'une image signifierait une plus grande importance ou non.

Le nombre de couleurs

La couleur de l'image pourrait aussi déterminer l'importance d'une image. Plus le nombre de couleurs est grand et plus l'image pourrait être considérée comme importante. Par exemple, une image qui est composée d'un grand nombre de pixels blancs ou noirs ou même transparents peut sembler moins importante qu'une autre.

Le format de l'image

Chaque image est caractérisée par un format. Ce dernier peut être JPEG, PNG ou même GIF. Chacun de ces formats possède des caractéristiques particulières comme le nombre de couleurs qui est limité à 256 pour le format GIF. On pourrait donc penser qu'une image ayant le format GIF est moins importante qu'une image JPEG.

Tous ces critères, ou tout du moins une partie, pourraient être étudiés afin de déterminer un vecteur de caractéristiques pour chacune des images de la page. Suivant ces caractéristiques, on déterminerait alors un poids pour chaque image.

Une fois tous les poids assignés, il faudrait corriger les images qui ont le poids le plus fort en priorité. Cette mise en place pourrait faire l'objet d'un projet à part, destiné à un étudiant de Polytech'Tours par exemple.

8.12 Rédaction du rapport de PFE

Évidemment, un rapport devait être rédigé.

Le livrable est un document de type PDF réalisé avec le système de composition de documents nommé Latex.

8.13 Préparation de la présentation orale finale

Une présentation orale est prévue le 6 Mai 2013 avec la présence des encadrants et des autres élèves travaillant au sein de l'équipe Handicap et Nouvelles Technologies.

Cette présentation orale va permettre de faire un bilan final du projet. Elle sera d'abord composée d'une présentation du projet puis le jury pourra poser les questions qu'il souhaite. Cette soutenance finale est primordiale puisqu'elle permet de rendre compte d'un travail ayant débuté fin Septembre 2012.

Analyse critique du projet

9.1 Difficultés rencontrées

Durant ce projet, de nombreuses difficultés ont été rencontrées. D'abord, l'analyse de l'existant a été une étape difficile du projet. En effet, la plateforme SWAP est composée d'un grand nombre de sous-projets, où chaque projet contient plus ou moins de classes. Il était donc assez compliqué de prendre en main le projet SWAP d'une part à cause de la taille conséquente du projet, et d'autre part à cause de la complexité de fonctionnement de la plateforme. Au final, la prise en main de la plateforme s'est faite au fur et à mesure que le projet avançait. Ceci s'explique car pour chaque nouvelle fonctionnalité que j'avais à développer, je devais m'intéresser à des classes existantes pour y parvenir.

La manipulation des images a été une étape assez compliquée à aborder. Auparavant, je n'avais jamais manipulé des images avec le langage Java, ce qui rendait la tâche plus ardue. De plus, la gestion de la transparence de certaines images a posé beaucoup de problèmes.

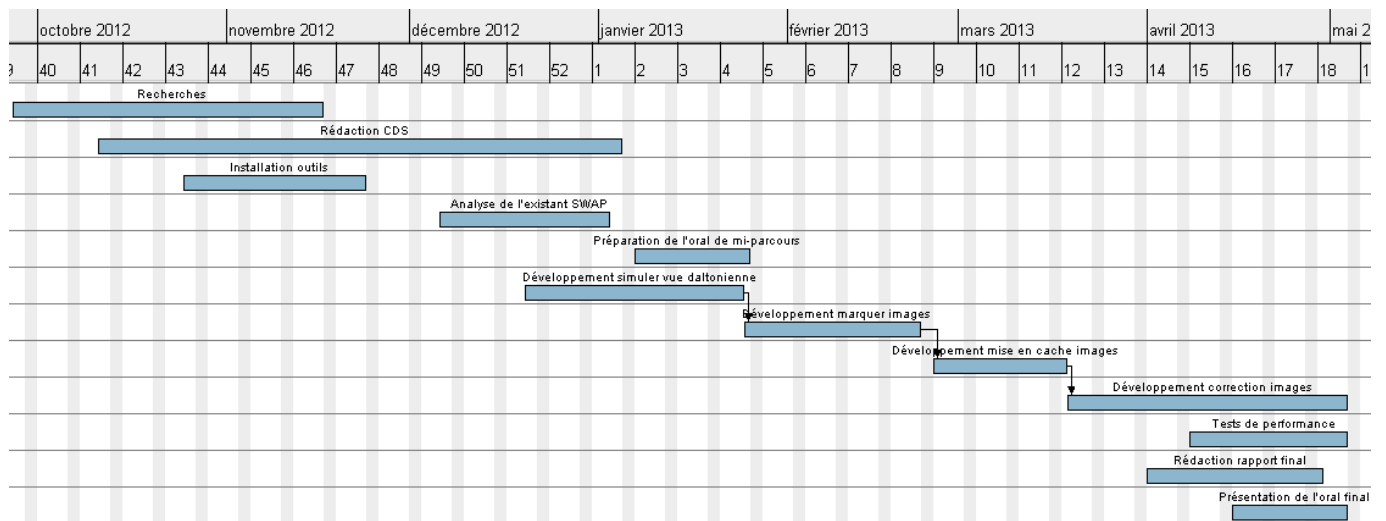
Enfin, la mise en place des algorithmes de correction pour compenser les pertes visuelles dues au daltonisme a également été une étape difficile. D'abord pour l'algorithme avec le passage dans l'espace de couleur LMS car les valeurs des matrices permettant de faire la correction sont rarement les mêmes sur le web. Ensuite, pour ce qui est de l'algorithme de correction avec le passage par l'espace de couleur CIE Lab, il y avait peu de sources disponibles et aucune implémentation n'a pu être trouvée. De plus, l'explication de cette méthode est faite de manière très théorique, ce qui complexifie l'implémentation.

Il est normal de rencontrer des problèmes dans un projet informatique, et c'est en les résolvant un par un que l'on avance et apprend.

Faire un bilan final en comparant le planning établi dans le cahier de spécifications et le planning final illustrant ce qui a réellement été fait est important.

octobre 2012				novembre 2012				décembre 2012				janvier 2013				février 2013				mars 2013				avril 2013				mai							
40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18					
Recherches																																			
Rédaction CDS																																			
Installation outils																																			
																		Analyse de l'existant SWAP																	
																		Préparation oral mi-parcours																	
																		Développement simulateur vue daltonienne																	
																		Développement et marquer images modifiées																	
																		Développement interceptor ensemble images HTML																	
																		Développement mise en cache images modifiées																	
																		Tests sur ensemble de l'application																	
																		Rédaction rapport final																	
																		Présentation oral final																	

Pour comparer le planning prévisionnel avec le travail réalisé, voici le planning effectif à la fin de ce projet :



On remarque que jusqu'à l'échéance de l'oral de mi-parcours, c'est-à-dire vers fin Janvier, le planning a été respecté. Ensuite, en ce qui concerne les tâches qui ont été réalisées à partir du mois de Février, on remarque que le planning a globalement été respecté même si la tâche "Développement intercepter l'ensemble des images HTML" avait été mal défini puisque cette tâche était en fait intégrée à la tâche suivante qui consistait à mettre en cache les images des pages voisines.

Par ailleurs, plusieurs tâches n'avaient pas été spécifiées lors de la rédaction du cahier de spécifications, comme la correction des images ou les tests de performance des différents algorithmes de correction. Il était difficile d'appréhender la durée de la tâche pour la correction des images. En effet, plusieurs algorithmes existent et il fallait les étudier puis les mettre en place. Le premier algorithme mis en place était la correction avec LMS(8.9.2). Cet algorithme a longtemps été testé afin d'obtenir un résultat convenable. Par la suite, la mise en place de la correction avec CIE Lab (8.9.3) a été étudiée mais pas implémentée par manque de temps. En effet, cette correction d'image est expliquée de façon très théorique et la compréhension de cette méthode est plutôt longue.

De plus, suite aux tests de performance des algorithmes de correction, une priorisation des images devait être effectuée. Le faible temps qui restait pour réaliser les différentes tâches citées précédemment a fait que la partie priorisation n'a pas pu être abordée.

Conclusion

Ce projet de fin d'études réalisé lors de ma cinquième année au département informatique de l'école Polytechnique de l'Université de Tours fut très intéressant. Le but principal de ce projet était de compenser sur les images les pertes visuelles dues au daltonisme. Avant de chercher à corriger les troubles visuels d'un daltonien, de nombreuses fonctionnalités ont été mises en place comme la simulation d'une vue daltonienne sur une page web ou le principe de mise en cache des images afin d'accélérer le chargement de celles-ci.

Durant ce projet, le planning prévisionnel qui avait été défini lors des spécifications a été globalement respecté. Cependant, des tâches de développement n'avaient pas été spécifiées au départ. Cela implique que ces développements qui ont été ajoutés à la fin du projet n'ont pas pu être tous terminés, comme la partie priorisation des images.

Ce projet fut enrichissant car il m'aura permis de m'affirmer en tant que futur ingénieur. Le fait de mener un projet seul sur une année entière aura été très bénéfique pour moi, car j'ai pu m'investir complètement dans la réalisation de ce projet. De plus, outre le fait de devoir gérer moi-même ce projet, il était très agréable de se sentir soutenu et encadré par Mr. Mohamed Slimane et Mr. Sébastien Aupetit. D'une part pour le fait de se tenir au courant de l'avancement du projet et d'autre part pour l'aide technique qui pouvait m'être apportée lorsque je rencontrais des problèmes.

Enfin, ce projet me tenait vraiment à cœur étant moi-même atteint de daltonisme. Il était donc aisé pour moi de tester l'implémentation des algorithmes permettant de compenser les pertes visuelles dues au daltonisme.

Bibliographie

- [1] Comment ça marche. Cie lab. <http://www.commentcamarche.net/contents/video/cie-lab.php3>.
- [2] Comment ça marche. Protocole http. <http://www.commentcamarche.net/contents/internet/http.php3>.
- [3] Blog couleur. Srgb. <http://www.blog-couleur.com/?Qu-est-ce-que-le-sRGB>.
- [4] Daltonisme. Le daltonisme. <http://www.daltonisme.com/>.
- [5] Chrome Daltonize. Simulation vue daltonienne. <https://chrome.google.com/webstore/detail/chrome-daltonize/efeladnkafmoofnbagdbfaieabmejfcf>.
- [6] Datoniens. Les types d'anomalies. <http://www.daltoniens.fr/>.
- [7] Frédéric Dufour. Les dix règles de jacob nielsen. <http://fr.slideshare.net/fredericdufour/lutlisabilit-dun-site-web>.
- [8] Manuel M. Oliveira Giovane R. Kuhn and Leandro A. F. Fernandes. An efficient naturalness-preserving image-recoloring method for dichromats. http://www.inf.ufrgs.br/~oliveira/pubs_files/CVD_Mass_Spring/CVD_Mass_Spring/Kuhn_Oliveira_Fernandes_Vis2008.pdf.
- [9] John D. Mollon Hans Brettel, Françoise Viénot. Computerized simulation of color appearance for dichromats. <http://vision.psychol.cam.ac.uk/jdmollon/papers/Dichromatsimulation.pdf>.
- [10] Marmalouka. Accessibilité. http://marmalouka.free.fr/site_accessibilite/info_w3c.html.
- [11] Sébastien Aupetit Mohamed Slimane. Smart web proxy for web accessibility. <http://hant.li.univ-tours.fr/webhant/index5fbc.html?pageid=67>.
- [12] Color Oracle. Simulation vue daltonienne. <http://colororacle.org/>.
- [13] Vischeck. Simulation vue daltonienne. <http://www.vischeck.com/>.
- [14] W3C. Introduction à l'accessibilité du web. <http://www.w3qc.org/ressources/traductions/introduction-accessibilite-web/?/docs/accessibilite/>.
- [15] Wikipédia. Espace colorimétrique. http://fr.wikipedia.org/wiki/Espace_colorimetrique.
- [16] Wikipédia. Rgb. http://fr.wikipedia.org/wiki/Red_Green_Blue.
- [17] Wikipedia. Lms color space. http://en.wikipedia.org/wiki/LMS_color_space.
- [18] William Woods. Modifying images for color blind viewers. http://www.stanford.edu/class/ee368/Project_12/Reports/Woods_Assisting_Color_Blind_Viewers.pdf.

Glossaire

A

API Une interface de programmation (en anglais Application Programming Interface ou API) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine., p. 21.

B

bâtonnets Cellule photo-réceptrice de la rétine, qui doit son nom à sa forme allongée. Les bâtonnets sont très sensibles à la quantité de lumière., p. 16.

C

CIE La Commission internationale de l'éclairage (CIE) est une organisation internationale dédiée à la lumière, l'éclairage, la couleur, les espaces de couleur. Cette commission a été créée pour caractériser rationnellement les couleurs des lumières telles que les voit le cerveau humain., p. 16.

cônes Cellules visuelles, situées essentiellement au centre de la rétine, et responsables de la vision des détails (l'acuité visuelle) et de la vision des couleurs., p. 11.

G

gamut Gamme de couleur, sous-ensemble complet de couleurs qu'un certain type de matériau ou d'écran permet de reproduire., p. 14.

H

HTML HyperText Markup Language. Langage permettant de créer des pages Web, il utilise une structure formée avec des balises permettant la mise en forme du texte., p. 13.

HTTP Protocole de communication utilisé pour l'accès à un serveur web sécurisé., p. 13.

I

IDE Integrated Development Environment en anglais et Environnement de Développement Intégré (EDI) en français. C'est un programme regroupant un ensemble d'outils pour le développement de logiciels., p. 19.

M

Maven Apache Maven est un outil logiciel libre pour la gestion et l'automatisation de production des projets logiciels Java en général., p. 19.

O

OpenCV OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel., p. 21.

P

proxy Mandataire en informatique (proxy en anglais) est un composant logiciel qui se place entre deux autres pour faciliter ou surveiller leurs échanges., p. 13.

T

thread Un thread est similaire à un processus car tous deux représentent l'exécution d'un ensemble d'instructions du langage machine d'un processeur. Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en parallèle., p. 24.

W

WAI Désigne un des quatre domaines du W3C et l'organisation affiliée au consortium ayant pour mission la promotion de l'accessibilité en matière de sites Internet. Sa principale vocation est de proposer des solutions pour rendre Internet accessible aux personnes handicapées., p. 8.

WCAG Web Content Accessibility Guidelines se développe à travers le processus du W3C en collaboration avec des personnes et des organisations du monde entier. Les documents WCAG expliquent comment rendre les contenus Web plus accessibles aux personnes handicapées., p. 8.

Annexes

A.1 Cahier de spécifications système



École Polytechnique de l'Université de Tours
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. +33 (0)2 47 36 14 14
www.polytech.univ-tours.fr

Département Informatique

Cahier de spécification système & plan de développement			
Projet :	Adaptation des images d'un site web pour la compensation du daltonisme		
Emetteur :	Pierre Fourreau	Coordonnées : EPU-DI	
Date d'émission :	28 avril 2013		
Validation			
Nom	Date	Valide (O/N)	Commentaires

Mohamed Slimane : 20/12/2012 ; O ; Commentaires faits sur le document par écrit

Sébastien Aupetit : 20/12/2012 ; O ; Commentaires faits sur le document par écrit

Historique des modifications		
Version	Date	Description de la modification

1.1 : 01/12/12 ; Version initiale

1.2 : 28/12/12 ; Modification de la version initiale (points à revoir)

Table des matières

Cahier de spécification système	6
1.1 Introduction	6
1.2 Contexte de la réalisation	6
1.2.1 Contexte	6
1.2.2 Objectifs	6
1.2.3 L'accessibilité sur le web	7
1.2.4 La notion d'utilisabilité	7
1.2.5 Le daltonisme	7
1.2.6 Bases méthodologiques	8
1.3 Description générale	8
1.3.1 Environnement du projet	8
1.3.2 Caractéristiques des utilisateurs	9
1.3.3 Fonctionnalités et structure générale du système	9
1.3.4 Contraintes de développement, d'exploitation et de maintenance	9
1.4 Description des interfaces externes du logiciel	10
1.5 Architecture générale du système	10
1.5.1 Composants centraux	10
1.5.2 Composants du proxy	11
1.5.3 Composants du serveur	11
1.6 Description des fonctionnalités	11
1.6.1 Définition de la fonction permettant de simuler une vue daltonienne	11
1.6.2 Définition de la fonction permettant de marquer les images modifiées	12
1.6.3 Définition de la fonction permettant d'intercepter toutes les images d'une page web	12
1.6.4 Définition de la fonction permettant de mettre en cache les images modifiées	12
1.6.5 Définition de la fonction permettant de mettre en cache les pages voisines	13
1.7 Conditions de fonctionnement	13
1.7.1 Performances	13
1.7.2 Méthodologie	13
1.7.3 Contrôlabilité	13
 Plan de développement	 15
2.1 Découpage du projet en tâches	15
2.1.1 Recherches et documentation	15
2.1.2 Installation de l'environnement de développement	15
2.1.3 Analyse de l'existant de la plateforme SWAP	15
2.1.4 Rédaction du cahier de spécifications	16
2.1.5 Préparation de la présentation orale à mi-parcours	16
2.1.6 Développement permettant de simuler une vue daltonienne	16
2.1.7 Développement permettant de marquer les images modifiées	17
2.1.8 Développement permettant d'intercepter toutes les images d'une page web	17
2.1.9 Développement permettant de mettre en cache les images modifiées	18
2.1.10 Tests sur l'ensemble de l'application	18



2.1.11	Rédaction du rapport final du projet de fin d'études	18
2.1.12	Préparation de la présentation orale finale	18
2.2	Planning	20
3	Glossaire	21

Cahier de spécification système

1.1 Introduction

Dans le cadre de notre formation au sein du Département Informatique de l'école Polytechnique de l'Université de Tours, un projet de fin d'études (PFE) est effectué afin de mettre en pratique les connaissances acquises au cours de notre formation.

Ce document a pour but de spécifier le projet de fin d'études intitulé : « adaptation des images d'un site web pour la compensation du daltonisme ».

Ce projet est réalisé au sein de l'équipe Handicap et Nouvelles Technologies (HaNT) de François Rabelais.

Ce cahier de spécifications est rédigé par moi-même, Pierre Fourreau. Les professeurs chargés de l'encadrement et de la relecture de ce projet sont Mohamed Slimane et Sébastien Aupetit.

1.2 Contexte de la réalisation

1.2.1 Contexte

Au départ, le web avait été mis en place pour pouvoir être utilisé par tout le monde. Mais il s'avère qu'aujourd'hui, cet objectif n'est pas atteint. Actuellement, de nombreux sites web ne sont pas ou peu accessibles/utilisables pour des personnes souffrants d'un handicap visuel (comme le daltonisme). Ainsi, des limitations fonctionnelles empêchent des millions de personnes d'utiliser le Web. Ceci peut être expliqué lorsque l'on sait que le développement d'un site web est soumis à un ensemble de contraintes (financières ou d'ergonomies). De plus, il n'est pas rare qu'un site web soit délaissé par celui qui a conçu ce site.[4]

Ici, le but est de rendre plus simple la navigation de l'utilisateur sur les sites web sans qu'il ait besoin de suivre un apprentissage spécifique. Les actions de l'utilisateur sur la page web doivent se faire de manière intuitive.

L'équipe HaNT de l'Université François Rabelais tente de mettre en place un outil facilitant l'accès au web pour les personnes handicapées. L'objectif de cet outil est de traiter les pages web afin de les rendre plus accessibles.

Mon projet de fin d'études est donc directement en relation avec l'outil mis en place par l'équipe HaNT puisque le sujet de mon projet est d'adapter des images d'un site web pour la compensation du daltonisme.

1.2.2 Objectifs

L'objectif est de permettre l'utilisation d'un web non accessible/non utilisable à des personnes ayant une déficience visuelle.

Dans ce projet, on cherche à compenser les pertes visuelles dues au daltonisme sur les images, à différencier les images du décor et les autres images, à prioriser les traitements dans le cas où le nombre d'images à traiter est trop importants pour le temps disponible. L'ensemble du projet sera réalisé en Java au sein de la plateforme SWAP (Smart Web Accessibility Proxy).

1.2.3 L'accessibilité sur le web

Dans le domaine du web, l'accessibilité signifie que les personnes handicapées peuvent utiliser le web. Cela permet à ces personnes de percevoir, comprendre, naviguer et interagir avec le Web ainsi que de pouvoir apporter leur contribution et créer du contenu. L'accessibilité du web vise toutes les déficiences, qu'elles soient visuelles, auditives, motrices, cognitives, neurologiques ou liées à la parole. Cependant, de nombreux sites web ne sont pas ou peu accessibles/utilisables pour des personnes souffrants d'un handicap visuel (comme le daltonisme). En effet, des limitations fonctionnelles empêchent des millions de personnes d'utiliser le Web.[6]

L'accessibilité peut être exigée par la loi. En effet, afin de limiter les problèmes et permettre un accès au web à tous, les associations, lobbys et autres législateurs demandent à ce qu'un certain nombre de principes et normes (WCAG1, WCAG2) soient respectés par les concepteurs de site web. Des lois existent et tentent d'obliger les concepteurs à concevoir des sites web accessibles. Malheureusement, ces lois, normes et principes ne sont pas suffisants.

Rendre un site web accessible à 100% reste utopique pour de nombreuses raisons. Cela serait trop coûteux en temps et argent, trop complexe et trop contraignant. En effet, beaucoup de sites ne sont plus maintenus mais existent toujours, ou d'autres sont faits manuellement sans tenir compte de la notion d'accessibilité. Par ailleurs, de nombreux sites web ne sont pas d'origine française ; les lois et les contraintes peuvent donc être différentes voire ne pas exister. Un site web peut être accessible à une personne ayant le handicap X sans l'être pour autant l'être avec une autre ayant le handicap Y. De ce fait, pour répondre au problème d'accessibilité sur des sites web, une solution peut consister à viser une meilleure accessibilité sur ces derniers plutôt que de viser une accessibilité complète.

1.2.4 La notion d'utilisabilité

Il faut bien faire la différence entre l'accessibilité d'un site web et l'utilisabilité de ce dernier. En effet, un site web peut très bien être accessible à une personne ayant une déficience visuelle mais cette personne peut être dans l'incapacité d'utiliser ce site. L'objectif de l'utilisabilité du web est de rendre la navigation sur les sites web plus aisée pour l'utilisateur, sans qu'il ait besoin de suivre un apprentissage spécifique. En effet, les actions de l'utilisateur sur la page web doivent se faire de manière intuitive.

Dans un site web, l'ergonomie correspond à répondre efficacement aux attentes de l'utilisateur afin de leur fournir une navigation plus aisée. Le problème est que les visiteurs du site web n'ont pas tous le même profil. Il faut donc prendre en considération les attentes de l'utilisateur, ses habitudes, son âge, ses équipements ainsi que son niveau de connaissance.[2]

1.2.5 Le daltonisme

Le projet concerne les personnes ayant une déficience visuelle et plus particulièrement les personnes atteintes de daltonisme.

Le daltonisme est une maladie qui est répandue dans le monde. Son apparition est due à une anomalie de la rétine. C'est un dysfonctionnement de la vision des couleurs, entraînant le plus souvent la confusion entre le rouge et le vert.[1] Le daltonisme est une anomalie dans laquelle un ou plusieurs des trois types de cônes de la rétine oculaire, responsables de la perception des couleurs sont déficients. Habituellement, le daltonisme est classé comme une infirmité légère. Cependant, il y a des situations où les daltoniens peuvent avoir un avantage sur les individus ayant une vision normale (percer certains camouflages basés sur la couleur).

Il existe plusieurs formes de daltonisme (aussi appelé dyschromatopsie) partielle, la plus fréquente étant la confusion du vert et du rouge. Les autres formes de daltonisme sont nettement plus rares, comme la confusion du bleu et du jaune, la plus rare de toutes étant la déficience totale de la perception des couleurs (achromatopsie), où le sujet ne perçoit que des nuances de gris.

Les différentes formes de daltonisme sont les suivantes :

- Achromatopsie
- Deutéranopie
- Deutéranomalie
- Protanopie
- Protanomalie
- Tritanopie
- Tritanomalie

Chacune de ces formes de daltonisme possède des caractéristiques particulières.[7]

1.2.6 Bases méthodologiques

Chaque semaine, un compte-rendu sera à rédiger par email et à envoyer aux encadrants. Dans chaque compte rendu, il faudra préciser les activités de la semaine, répondre aux questions ouvertes par le précédent compte-rendu, poser de nouvelles questions en indiquant le destinataire de la question, et enfin faire un rétroplanning de ce qui est à faire dans la semaine suivante.

La rédaction d'un compte rendu hebdomadaire présente un réel intérêt. En effet, cela permet de se rendre compte rapidement si le projet n'avance pas suffisamment, et de pouvoir retrouver facilement tout ce qui a été fait précédemment. De plus, ces comptes-rendus permettent aux encadrants de mieux suivre l'avancement du projet, et peuvent aussi permettre de discuter et d'échanger avec les encadrants.

Tous les documents relatifs au projet sont déposés sur un serveur Subversion qui est un logiciel de gestion de versions. Ces documents peuvent être des rapports, des comptes rendus ou même des codes sources.

L'avantage principal d'utiliser un serveur Subversion pour stocker tous ces documents est que l'on possède toutes les différentes versions de développement. Il est alors très aisé de revenir à un état antérieur de la phase de développement grâce à un logiciel client de SVN installé sur ma propre machine comme TortoiseSVN.[5]

De plus, l'autre avantage d'utiliser un serveur Subversion pour stocker tous ces documents est qu'en cas de problème, comme une perte des données, suite à un plantage d'une machine par exemple, tous les documents relatifs au projet pourront alors être récupérés grâce au serveur.

Le langage utilisé pour le développement de ce projet sera le Java 1.7.

1.3 Description générale

1.3.1 Environnement du projet

Ce projet s'insère dans le projet SWAP (Smart Web Accessibility Proxy). SWAP est un ensemble de classes Java permettant le développement rapide d'un proxy. Le proxy SWAP a des applications spéciales dans l'amélioration de l'accessibilité du site Web. Il est capable de transformer les pages web non accessibles en page web plus accessibles.

1.3.2 Caractéristiques des utilisateurs

Il est assez facile de caractériser les utilisateurs. En effet, l'objectif de ce projet est de permettre l'utilisation d'un site web non accessible/non utilisable à des personnes ayant une déficience visuelle. Plus particulièrement, l'utilisateur sera daltonien.

Cet utilisateur n'a pas nécessairement besoin de connaître l'informatique mis à part de savoir utiliser un navigateur internet et naviguer sur des sites web.

1.3.3 Fonctionnalités et structure générale du système

L'équipe HaNT a développé et mis en place un système de proxy appelé SWAP. Cet acronyme signifie Smart Web Accessibility Proxy. Ce proxy est codé en Java et doit donc être présent sur la machine de l'utilisateur ayant une déficience visuelle. Ce système permet de récupérer une page web et d'effectuer des traitements sur cette même page pour ensuite pouvoir l'afficher à l'utilisateur.

L'intérêt de ce système est de concevoir et mettre en place un proxy HTTP intelligent capable de transformer les pages web non accessibles en page web plus accessibles. Grâce à l'utilisation d'un proxy HTTP, il est possible d'intervenir directement sur le document et sa forme sans pour autant changer les habitudes et les logiciels spécifiques de l'utilisateur comme le lecteur braille ou la synthèse vocale. De ce fait, le proxy est présent pour faire bénéficier l'utilisateur d'améliorations.

Dans le système, le proxy a une place importante. En effet, lorsqu'un utilisateur envoie une requête (demande une page web), le proxy va interagir avec le navigateur web sans que l'utilisateur s'en rende compte. Le proxy va alors se charger de récupérer et de traiter les données du navigateur pour finalement renvoyer une réponse à l'utilisateur (la page web demandée).

L'avantage d'un tel système est que le traitement réalisé par le proxy est invisible pour l'utilisateur.

De plus, des fonctionnalités sont déjà implémentées sur ce proxy. Il faudra donc en partie utiliser les fonctionnalités existantes afin de pouvoir permettre à une personne daltonienne de visualiser une page web comme un utilisateur n'ayant pas de problème de vision.

1.3.4 Contraintes de développement, d'exploitation et de maintenance

Contraintes de développement

L'ensemble du projet sera réalisé en Java au sein de la plateforme SWAP. Ce langage est évidemment choisi puisque le proxy a été développé avec ce langage.

Le projet doit être réalisé de préférence sur une machine LINUX.

Cette machine LINUX doit posséder :

- le kit de développement JDK 1.7
- le logiciel de gestion de versions Apache Subversion 1.7
- le logiciel pour la gestion et l'automatisation de production des projets Java Apache Maven 3

Le développement se fera avec l'IDE Eclipse version 4.2. Ce dernier doit posséder le plugin Subversive permettant l'utilisation de Subversion et le plugin Maven Integration (m2e).

Maintenance et évolution du système

La maintenance sera assurée par le logiciel de gestion de versions Apache Subversion (SVN).

1.4 Description des interfaces externes du logiciel

Comme expliqué précédemment, le proxy développé par le laboratoire HaNT permet de récupérer une page web et d'effectuer des traitements sur cette même page pour ensuite pouvoir l'afficher à l'utilisateur. Il y a donc une liaison entre le navigateur web de l'utilisateur et le proxy.

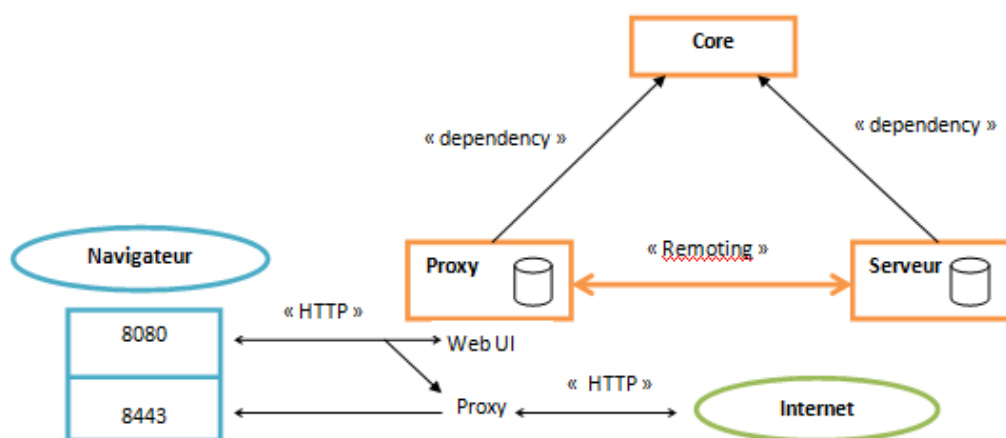


FIGURE 1.1 – Architecture du système

Dans ce schéma on voit que le composant Core est un composant commun du proxy et du serveur. Ces deux derniers peuvent communiquer grâce au remoting (appel de procédure distante RPC).

L'intérêt de ce schéma est de montrer que le navigateur peut accéder à une page web via le proxy. Dans ce cas ce sont les ports HTTP (8080) et HTTPS (8443) qui sont utilisés. Mais ce schéma montre aussi que le navigateur peut accéder à une page sans que le proxy ait à accéder à internet. En effet, le Web UI est un composant du proxy, et dans le cas où la page demandée par le navigateur existe déjà sur le proxy, le Web UI se charge alors de renvoyer la page demandée.

1.5 Architecture générale du système

La plateforme SWAP possède de multiples composants. Ces derniers peuvent être regroupés en trois grandes parties. Il y a les composants centraux (core components), les composants du proxy, et ceux du serveur. [3]

1.5.1 Composants centraux

Pour les composants centraux (core components), différentes classes (content helper classes) existent comme les classes HTTP, les classes d'encodage, les classes HTML et les classes CSS.

Les composants de SWAP sont centrés sur un environnement objet. Les objets managent le cycle de vie de l'application, l'accès à la configuration et le contexte de l'application. La plupart des composants de SWAP reçoivent une référence sur l'interface Environment. Il y a 2 environnements concrets : ProxyEnvironment and ServerEnvironment. Ces deux derniers représentent une application proxy ou une application serveur.

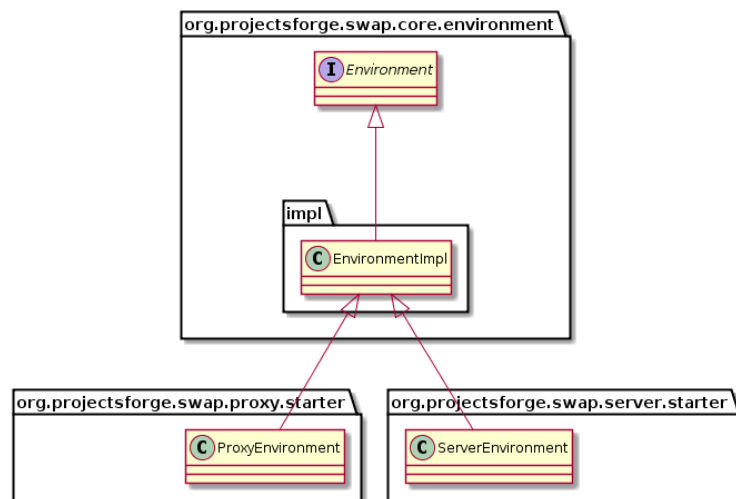


FIGURE 1.2 – Environnement SWAP

Lorsque l'on a un traitement à faire et que l'on a plusieurs tâches. Si on a plusieurs classes qui sont déclarées en tant qu'handler, on va pour gérer les exécutions parallèles de l'ensemble des tâches.

Les handlers sont aussi capables de se partager des ressources.

D'autre part, des classes permettent de manipuler le contenu des pages web (content helper classes). Il y a des classes permettant de construire une requête HTTP et de manipuler le cache (HTTP helper classes) ou alors de calculer l'encodage du document pour une manipulation correcte du contenu (encoding helper classes). Enfin, d'autres classes existent et permettent notamment de manipuler le HTML et le CSS.

1.5.2 Composants du proxy

Pour les composants du proxy, les gestionnaires de contenu (content handlers) sont composés des gestionnaires MIME, HTML, CSS et des gestionnaires d'images. Ensuite, il y a plusieurs servlets pour la partie proxy avec une servlet pour la commutation, pour le proxy et pour le WebUI.

1.5.3 Composants du serveur

Pour les composants du serveur, il existe aussi plusieurs servlets notamment pour le remoting et pour le WebUI.

1.6 Description des fonctionnalités

1.6.1 Définition de la fonction permettant de simuler une vue daltonienne

Identification de la fonction permettant de simuler une vue daltonienne

Le nom de cette fonction sera SimulationDaltonisme. Son rôle sera de simuler les trois types de daltonisme des images présentes sur le web. Il y a donc la vision deutéranope, protanope et tritanope à simuler. Avant de pour effectuer cette simulation, il faudra intercepter toutes les images présentes sur le page web. Afin d'intercepter les images, un handler est déjà mis en place sur la plateforme SWAP et se charge d'intercepter les images et de donner en entrée un flux de bytes correspondant à l'image interceptée.

La priorité de cette fonction est primordiale car elle constitue la base du traitement à effectuer.

Description de la fonction permettant de simuler une vue daltonienne

En entrée, nous auront une page HTML avec l'ensemble des balises possibles. Le handler qui intercepte les images fournit un flux de bytes correspondant à l'image interceptée.

1.6.2 Définition de la fonction permettant de marquer les images modifiées

Identification de la fonction permettant de marquer les images modifiées

Le nom de cette fonction sera MarquerImage. Son rôle sera d'ajouter une marque aux images qui ont été modifiées par le proxy avec la simulation de la vue daltonienne. Comme pour la fonction précédente, avant de pour effectuer cette simulation, il faudra intercepter toutes les images présentes sur le page web. Afin d'intercepter les images, un handler est déjà mis en place sur la plateforme SWAP et se charge d'intercepter les images et de donner en entrée un flux de bytes correspondant à l'image interceptée.

Cette fonction permettrait de savoir quelles images ont été simulées ou non sur la page web.

Description de la fonction permettant de marquer les images modifiées

En entrée, nous auront une page HTML avec l'ensemble des balises possibles. Le handler qui intercepte les images fournit un flux de bytes correspondant à l'image interceptée. Le handler fournit aussi diverses informations sur la page web et sur l'image.

1.6.3 Définition de la fonction permettant d'intercepter toutes les images d'une page web

Identification de la fonction permettant d'intercepter toutes les images d'une page web

Le nom de cette fonction sera InterceptorHTML. Son rôle sera d'intercepter toutes les images présentes sur une page web. Afin d'intercepter les images, un handler est déjà mis en place sur la plateforme SWAP, ce handler gère une image à la fois. De plus, un handler HTML existe pour gérer le HTML et un autre handler gère le CSS. Il faut donc détecter les images dans le HTML et dans le CSS et faire du prefetch (mise en cache) des images référencées.

Description de la fonction permettant d'intercepter toutes les images d'une page web

Il faut détecter les images dans le handler HTML puis les récupérer de manière asynchrone. Lorsque que le navigateur analysera la page HTML, il détectera les images et les demandera une par une au proxy. Si le proxy les a déjà en cache alors elles seront retournées tout de suite. Enfin, un marquage des requêtes pourra être effectué. Ceci sera effectué grâce à la marque qui aura été précédemment effectué sur les images.

1.6.4 Définition de la fonction permettant de mettre en cache les images modifiées

Identification de la fonction permettant de mettre en cache les images modifiées

Le nom de cette fonction sera MiseEnCacheImagesModifiees. Son rôle sera de mettre en cache les images qui auront été modifiées.

Description de la fonction permettant de mettre en cache les images modifiées

Cette fonctionnalité fait suite aux précédentes puisqu'elle prend en entrée les images modifiées. Une image modifiée est une image qui a été traitée par le proxy. Cette image peut être modifiée pour simuler une vision daltonienne.

1.6.5 Définition de la fonction permettant de mettre en cache les pages voisines

Identification de la fonction permettant de mettre en cache les pages voisines

Le nom de cette fonction sera `MiseEnCachePagesVoisines`. Son rôle sera de mettre en cache les pages voisines de la page courante dans le navigateur.

Description de la fonction permettant de mettre en cache les pages voisines

Cette fonctionnalité consiste à faire du calcul anticipé. Cette fonction devra télécharger les pages HTML (via un objet nommé `Request`).

1.7 Conditions de fonctionnement

1.7.1 Performances

Le proxy étant un intermédiaire entre un utilisateur et internet, il va falloir faire en sorte de retourner des résultats en un temps limité et raisonnable. Le but est de faire le mieux possible pendant ce temps limité.

1.7.2 Méthodologie

Chaque semaine, un compte-rendu indiquant les activités réalisées pendant la semaine est rédigé et envoyé aux encadrants de ce PFE. De plus, nous utilisons un SVN afin de garder une trace de l'avancement de notre projet. Enfin, différents comptes-rendus seront rédigés suite à des réunions effectuées avec les encadrants de ce PFE.

1.7.3 Contrôlabilité

Pour suivre l'exécution des traitements, il faudra utiliser des fichiers de log. En effet, dans ce projet, des loggers seront accessibles.

Plan de développement

2.1 Découpage du projet en tâches

Cette partie du cahier de spécifications permet de structurer le projet en tâches. Une tâche correspond ici à un ensemble de réalisations ayant une cohésion d'ensemble.

2.1.1 Recherches et documentation

Description de la tâche

Tout d'abord, le projet commence par une longue phase de recherches et de documentation. Cela consiste à se documenter sur tout ce qui tourne au daltonisme, comprendre comment fonctionne cette déficience visuelle. Il faut d'abord étudier la notion d'accessibilité et d'utilisabilité sur le web, puis trouver comment modifier les couleurs d'une image afin de passer de la vue d'une « personne normale » à une personne daltonienne. Il faudra aussi comprendre comment une couleur est représentée au niveau du codage (RGB par exemple), mais aussi de découvrir les différents algorithmes existants permettant de simuler la vue d'une personne daltonienne.

Livrables

Une synthèse résumant toutes les recherches doit être faite. Cette synthèse sera utile pour la rédaction du rapport final de ce projet. De plus, l'intérêt de cette synthèse est de réaliser une bibliographie avec tous les liens et documents qui ont été utilisés pour ces recherches. L'intérêt de cette bibliographie est de pouvoir retrouver facilement les différents liens qui m'auront permis de comprendre mon projet de fin d'études.

Estimation de charge

Cette tâche est estimée à 8 jours/homme.

2.1.2 Installation de l'environnement de développement

Description de la tâche

Cette tâche consiste à installer tous les outils nécessaires au développement sur la plateforme SWAP. Différentes installations doivent être effectuées comme l'installation de l'IDE Eclipse ainsi qu'un logiciel de gestion de version comme SVN. Un kit de développement JDK 1.7. De plus, plusieurs plugins sont à installer avec Eclipse comme Subversive et Maven 3.

Par ailleurs, il faudra récupérer le projet SWAP sur le serveur mis à disposition par nos encadrants. Pour cela il faudra utiliser le logiciel de gestion de versions SVN.

Estimation de charge

Cette tâche est estimée à 2 jours/homme.

2.1.3 Analyse de l'existant de la plateforme SWAP

Description de la tâche

Cette tâche consiste à prendre en main le code source existant du projet SWAP. Ce dernier étant composé de multiples classes, il faut pouvoir comprendre comment est structuré ce projet ainsi que savoir comment fonctionnent toutes ces classes.



Livrables

Une synthèse expliquant le fonctionnement général de la plateforme SWAP pourra être rédigée. Cette synthèse indiquera comment les différents composants de la plateforme SWAP interagissent entre eux.

Estimation de charge

Cette tâche est estimée à 4 jours/homme.

2.1.4 Rédaction du cahier de spécifications

Description de la tâche

Dans ce projet, un cahier de spécifications doit être rédigé. Ce document permet de décrire le projet d'un point de vue général d'abord, puis de spécifier l'architecture du système, les différentes fonctionnalités et de découper le projet en tâches.

Livrables

Le livrable est un document de type PDF réalisé avec le système de composition de documents nommé Latex.

Estimation de charge

Cette tâche est estimée à 5 jours/homme.

2.1.5 Préparation de la présentation orale à mi-parcours

Description de la tâche

Cette tâche consiste à préparer une présentation orale qui aura lieu au mois de Janvier. Il s'agit aussi de préparer un poster sur ce projet.

Livrables

Le livrable correspond ici à un rapport PDF du cahier de spécifications ainsi que d'un poster.

Estimation de charge

Cette tâche est estimée à 3 jours/homme.

2.1.6 Développement permettant de simuler une vue daltonienne

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de simuler une vue daltonienne.

1.6.1

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués, comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.7 Développement permettant de marquer les images modifiées

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de marquer les images modifiées.

1.6.2

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués, comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.8 Développement permettant d'intercepter toutes les images d'une page web

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant d'intercepter toutes les images du contenu HTML et CSS ainsi que de mettre en cache ces images.

1.6.3

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 12 jours/homme.

2.1.9 Développement permettant de mettre en cache les images modifiées

Description de la tâche

Cette tâche consiste à développer en Java au sein de la plateforme SWAP la fonction permettant de mettre en cache les images modifiées.

1.6.4

Cycle de vie

Des tests devront être effectués au fur et à mesure de l'avancement du développement. Différents types de tests pourront être effectués comme des tests unitaires, fonctionnels et d'intégration.

Livrables

Le livrable correspond ici au code source développé.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.10 Tests sur l'ensemble de l'application

Description de la tâche

Cette tâche consiste à tester l'ensemble de l'application qui comprend les différentes fonctions développées préalablement.

1.6.5

Cycle de vie

Différents types de tests pourront être effectués comme des tests unitaires, fonctionnels et d'intégration.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.1.11 Rédaction du rapport final du projet de fin d'études

Description de la tâche

Cette tâche consiste à rédiger le rapport final de ce projet de fin d'études.

Livrables

Le livrable correspond ici à un rapport PDF.

Estimation de charge

Cette tâche est estimée à 15 jours/homme.

2.1.12 Préparation de la présentation orale finale

Description de la tâche

Cette tâche consiste à préparer une présentation orale finale qui aura lieu le 3, le 6 et le 7 Mai 2013.

Livrables

Le livrable correspond ici à un rapport PDF final du projet de fin d'études.

Estimation de charge

Cette tâche est estimée à 6 jours/homme.

2.2 Planning

Le diagramme de Gantt correspondant aux tâches citées précédemment est le suivant :

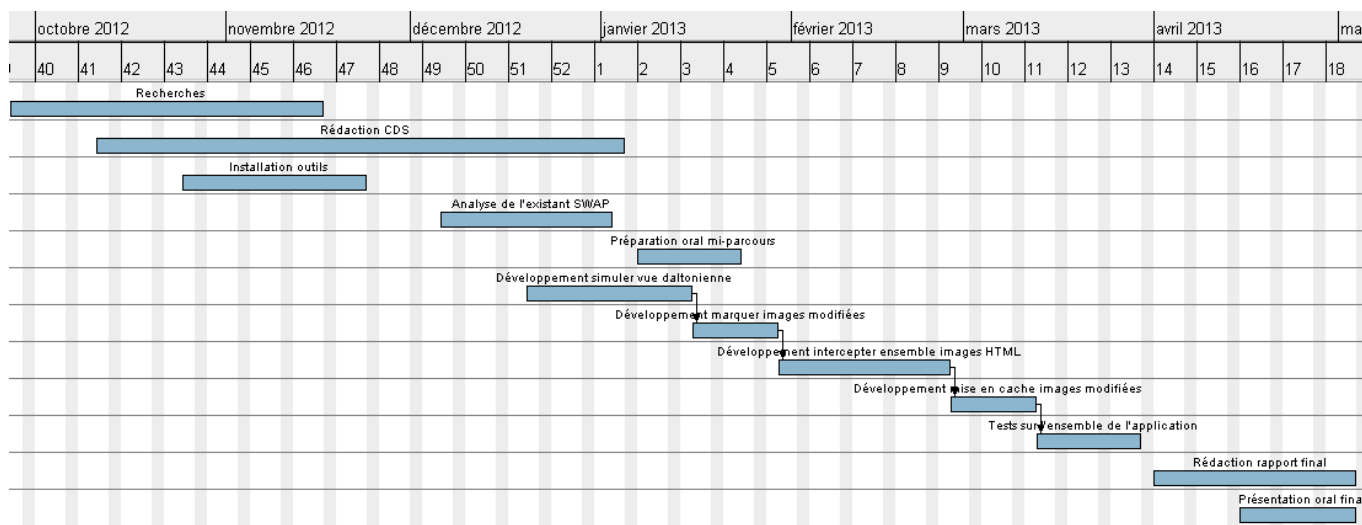


FIGURE 2.3 – Diagramme de Gantt

Il faut tenir compte du fait que le nombre de jours travaillés par semaine dans l'année peut être différent. En effet, de Septembre à fin Décembre, une seule journée par semaine est consacrée au projet de fin d'études, tandis qu'à partir du mois de Janvier et jusqu'à la fin du projet, trois journées y sont consacrées. Cela revient donc à un jour/homme par semaine sur le premier semestre et à 3 jours/homme sur le deuxième semestre.

Glossaire

Achromatopsie

Pathologie de la vision qui correspond à la non-perception des couleurs. Le patient achromate ne peut distinguer que le blanc, le noir, et les nuances de gris.

Dyschromatopsie

Défaut de la vision, caractérisé par une déficience partielle ou totale dans la distinction des couleurs (synonyme de daltonisme).

HTML

HyperText Markup Language. Langage permettant de créer des pages Web, il utilise une structure formée avec des balises permettant la mise en forme du texte.

HTTP

Protocole de communication utilisé pour l'accès à un serveur web sécurisé.

HTTPS

HyperText Transfer Protocol Secure est la combinaison du HTTP avec une couche de chiffrement comme SSL ou TLS.

IDE

Integrated Development Environment en anglais et Environnement de Développement Intégré (EDI) en français. C'est un programme regroupant un ensemble d'outils pour le développement de logiciels.

JDK

Java Development Kit. Logiciel édité par Sun pour le développement d'application en Java.

Proxy

Mandataire en informatique (proxy en anglais) est un composant logiciel qui se place entre deux autres pour faciliter ou surveiller leurs échanges.

RGB

Red Green Blue. Mode de description des couleurs comme mélange de trois composantes de bases, le rouge, le vert et le bleu.

Servlet

Une servlet est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP.

WCAG

Web Content Accessibility Guidelines se développe à travers le processus du W3C en collaboration avec des personnes et des organisations du monde entier. Les documents WCAG expliquent comment rendre les contenus Web plus accessibles aux personnes handicapées.

Bibliographie

- [1] Daltonisme. Le daltonisme. <http://www.daltonisme.com/>.
- [2] Frédéric Dufour. Les dix règles de jacob nielsen. <http://fr.slideshare.net/fredericdufour/lutlisabilit-dun-site-web>.
- [3] Projects Forge. Documentation swap. <https://projectsforge.org/projects/swap/wiki/doc/Index>.
- [4] Sébastien Aupetit Mohamed Slimane. Smart web proxy for web accessiblity. <http://hant.li.univ-tours.fr/webhant/index5fbc.html?pageid=67>.
- [5] Dev nozav. Subversion. http://dev.nozav.org/intro_svn.html.
- [6] W3C. Introduction à l'accessibilité du web. <http://www.w3qc.org/ressources/traductions/introduction-accessibilite-web/?/docs/accessibilite/>.
- [7] Wikipédia. Le daltonisme. <http://fr.wikipedia.org/wiki/Daltonisme>.

Adaptation des images d'un site web pour la compensation du daltonisme

Département Informatique

5^e année
2012 - 2013

Rapport de projet de fin d'études

Résumé : Ce projet de fin d'études concerne l'adaptation des images d'un site web pour la compensation du daltonisme. Ce projet s'insère dans le projet SWAP (Smart Web Accessibility Proxy) dont l'objectif est de permettre l'utilisation d'un web non accessible/non utilisable à des personnes ayant une déficience visuelle.

Mots clefs : Daltonisme - Site Web - Accessibilité - Utilisabilité - Simulation - Correction - Rapport

Abstract: This project deals with the adaptation of images of a website for the compensation of color blindness. This project is included in the SWAP (Smart Web Accessibility Proxy) project, whose objective is to allow the use of a non accessible/non usable web by people with visual impairments.

Keywords: ColorbBlindness - Website - Accessibility - Usability - Simulation - Correction - Report

Encadrants

Mohamed Slimane

mohamed.slimane@univ-tours.fr

Sebastien Aupetit

aupetit@univ-tours.fr

Etudiant

Pierre FOURREAU

pierre.fourreau@etu.univ-tours.fr

DI5 2012 - 2013

Université François-Rabelais, Tours