

FRANÇOIS Sébastien

Groupe 410

Jury n°1

Enseignant Tuteur :

M. FERRON C.

Université d'Angers



INSTITUT UNIVERSITAIRE  
DE TECHNOLOGIE

DUT Génie Électrique et Informatique  
Industrielle

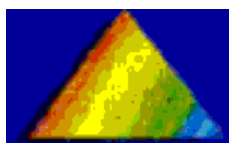
*Promotion 2003/2004*

## Étude d'une transposition de PortaNum™ sous environnement GNU/Linux

(du 13 Avril au 18 Juin 2004)



<http://seb.france.free.fr/iut/stage>



Institut Montclair  
51 rue du Vallon  
49000 Angers

FRANÇOIS Sébastien

Groupe 410

Jury n°1

Enseignant Tuteur :

M. FERRON C.

Université d'Angers



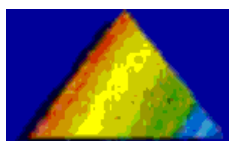
INSTITUT UNIVERSITAIRE  
DE TECHNOLOGIE

DUT Génie Électrique et Informatique  
Industrielle

*Promotion 2003/2004*

## Étude d'une transposition de PortaNum™ sous environnement GNU/Linux

Conduite du projet



Institut Montclair  
51 rue du Vallon  
49000 Angers

## Remerciements

*Je tiens à remercier M. Francis GUITTEAU, Directeur de l'Institut Montclair, et M. Jérôme SURUT Ingénieur Recherche et Développement, de m'avoir permis d'effectuer ce stage au sein de l'Institut Montclair.*

*Je remercie tout particulièrement M. Jérôme SURUT pour ses conseils et la confiance qu'il m'a accordée dans le cadre de mon travail, tout en m'accueillant chaleureusement.*

*Je remercie Joseph COLINEAU et Loïc GOASDUFF pour le temps qu'ils ont bien voulu nous consacrer pour un cours de traitement d'image au sein de Thalès Recherche et Technologie, Orsay.*

*Je tiens aussi à remercier M. Sébastien GUEGNIARD, assistant de M. SURUT et Mme Monica BADIA, stagiaire, qui ont contribué à la bonne ambiance qui régnait dans le service.*

*Je remercie l'ensemble du personnel pour sa sympathie et le temps qu'il nous a consacré pour nous expliquer leur travail et la mission des différents services.*

*Enfin, je tiens à remercier les personnes de la communauté du logiciel libre que j'ai contactées dans le cadre de ce projet ou bien qui ont participé aux développements des composants que j'ai employés.*

# Sommaire

<b>Remerciements</b>	<b>3</b>
<b>La conduite du projet</b>	<b>6</b>
<b>1. Origines du projet et ses objectifs</b>	<b>6</b>
1.1. Origines	6
1.2. Contexte du projet	7
1.3. La situation existante	8
1.4. Les objectifs à atteindre	8
1.4.1. Cahier des charges de Giris	8
1.5. Les contraintes :	8
1.5.1. Matérielles	8
1.5.2. Méthodologiques	9
1.6. L'évaluation du résultat	9
<b>2. Les prévisions</b>	<b>9</b>
2.1. Méthodes et ressources	9
2.1.1. Examen du cahier des charges	9
2.2. Planning prévisionnel	10
<b>3. La réalisation</b>	<b>11</b>
3.1. Présentation générale	11
3.1.1. Planning réel	12
3.2. Présentation détaillée des méthodes de travail	13
3.2.1. Installation de l'environnement	13
3.2.2. Choix des outils de développement	14
3.2.3. La modélisation UML	14
3.2.4. Portabilité	15
3.2.5. Structuration du logiciel / Documentation	16
3.2.6. Mise au point	16
<b>4. Le Bilan</b>	<b>16</b>
4.1. Résultats obtenus	16
4.2. Bilan de l'ensemble du projet	17
<b>Glossaire</b>	<b>18</b>
<b>Bibliographie</b>	<b>19</b>

<b>Dimension professionnelle du stage et vécu</b>	<b>21</b>
1. Dimension professionnelle	21
1.1. Historique	21
1.2. Domaines de compétences de l'Institut	21
1.3. Service recherche et développement	23
1.4. Contacts avec les autres services	23
1.5. Contact avec l'extérieur	24
2. Vécu du stage	24
3. Mentions légales	25
4.a. Résumé du projet	26
4.b. Project Abstract	26

## La conduite du projet

### 1. Origines du projet et ses objectifs

#### 1.1. Origines

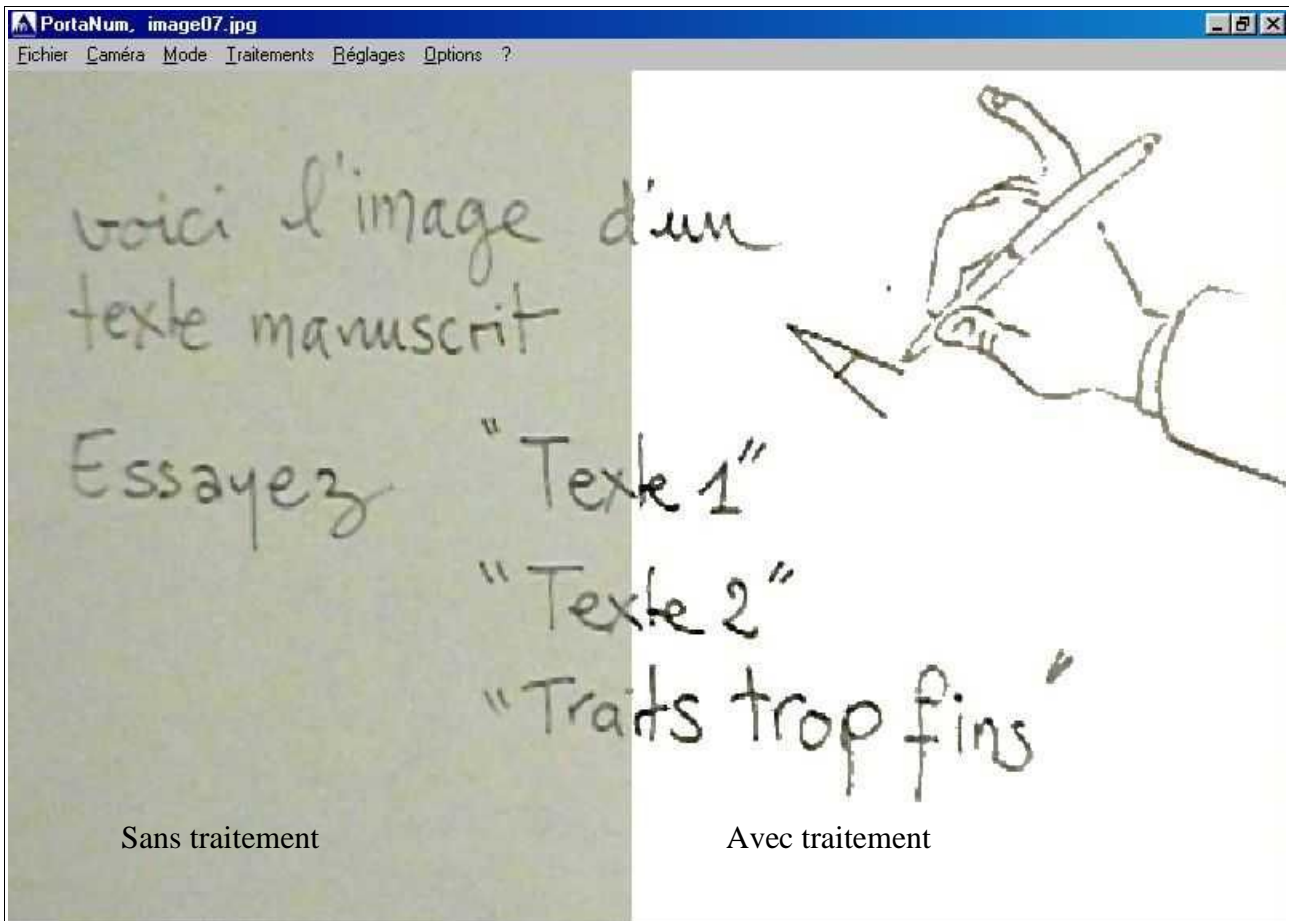
En février 2002, j'ai commencé à m'intéresser au problème de l'utilisation de GNU\*/Linux par des déficients visuels. J'ai participé à différents projets et ai créé, avec un ami, un système permettant à un non-voyant d'installer une distribution Debian de manière autonome et sans partitionnement du disque dur. Ces années de collaboration m'ont permis d'apprécier ce genre de travail et je souhaitais, si possible, réaliser un stage dans ce domaine. J'ai donc postulé auprès de l'Institut MONTÉCLAIR.

M. SURUT m'a expliqué que l'Institut dispose d'outils spécialisés permettant l'agrandissement, et que ceux-ci sont volumineux et coûteux. L'informatique étant déjà omniprésente dans le domaine de l'accessibilité aux déficients visuels, cette tâche peut être confiée à un ordinateur muni d'une caméra.



*Illustration 1 : Agrandisseur autonome*

La Société THALÈS a confié à Joseph COLINEAU et Hélène SOUBARAS l'étude et la réalisation d'un logiciel gratuit : PortaNum. Celui-ci permet d'appliquer un éventail de traitements numériques sur l'image provenant d'une webcam ou d'un camescope numérique. Mon stage consiste à étudier l'adaptation de ce logiciel sous environnement GNU\*/Linux.



Aperçu 1 : Portanum en action

## 1.2. Contexte du projet

Le stage s'effectue au sein du service informatique qui est responsable non seulement de la gestion du parc informatique de l'Institut (~70 machines) mais aussi du développement de logiciels pour l'évaluation et la rééducation visuelle.

Ce stage a pour but d'étudier la transposition sous GNU\*/Linux d'un logiciel d'aide à la vue, permettant par exemple de rendre lisible, à une personne déficiente visuelle, un cours sur tableau.

Le nom PortaNum étant déposé par la Société THALÈS, j'ai choisi de nommer le projet Giris, **G** car il s'appuie sur l'environnement de bureau **G**nome\* et **iris** parce que l'on associe immédiatement ce mot à la vue et qu'il est assez universel.

### 1.3. La situation existante

A l'heure actuelle, PortaNum n'est disponible que pour un environnement Microsoft Windows, il existe des projets de traitements d'images en temps réel sous Linux mais leur but n'est qu'artistique. J'ai étudié en détails PortaNum au cours de la préparation du stage (cf Dossier technique Étude de PortaNum).

### 1.4. Les objectifs à atteindre

#### 1.4.1. Cahier des charges de Giris

La conception du logiciel se limitera tout d'abord à résoudre un problème simple :

1. Acquérir l'image en continu à partir d'un périphérique vidéo
2. Avoir accès aux réglages matériels (luminosité, contraste, saturation etc...)
3. Afficher ensuite l'image traitée à l'écran de manière optimisée
4. Conserver l'ensemble des raccourcis clavier.

### 1.5. Les contraintes :

#### 1.5.1. Matérielles

Le développement s'effectuera parallèlement sur PC et sur Macintosh (PowerPC), et sera conforme aux standards de développement sous UNIX, pour permettre sa compilation sur d'autres architectures.

Le logiciel devra supporter les périphériques suivants :



La caméra USB **ToUcam pro II** de Philips

(On peut changer l'objectif afin de rendre la caméra plus sélective, l'image sera donc déjà fortement agrandie)



Le camescope numérique  
**VMD 912** de Thomson



### 1.5.2. Méthodologiques

M. SURUT a souhaité que j'utilise une modélisation en **UML\***, conformément aux habitudes du service. Les informations que je serai amené à rencontrer seront regroupées sous forme de petites études, pour permettre un accès plus aisé aux spécificités de l'environnement, et pour donner un aperçu du contexte du projet.

### 1.6. L'évaluation du résultat

Elle sera effectuée en fin de stage par M. SURUT, et le résultat pourra aussi être présenté à la personne s'occupant de la formation informatique des étudiants.

## 2. Les prévisions

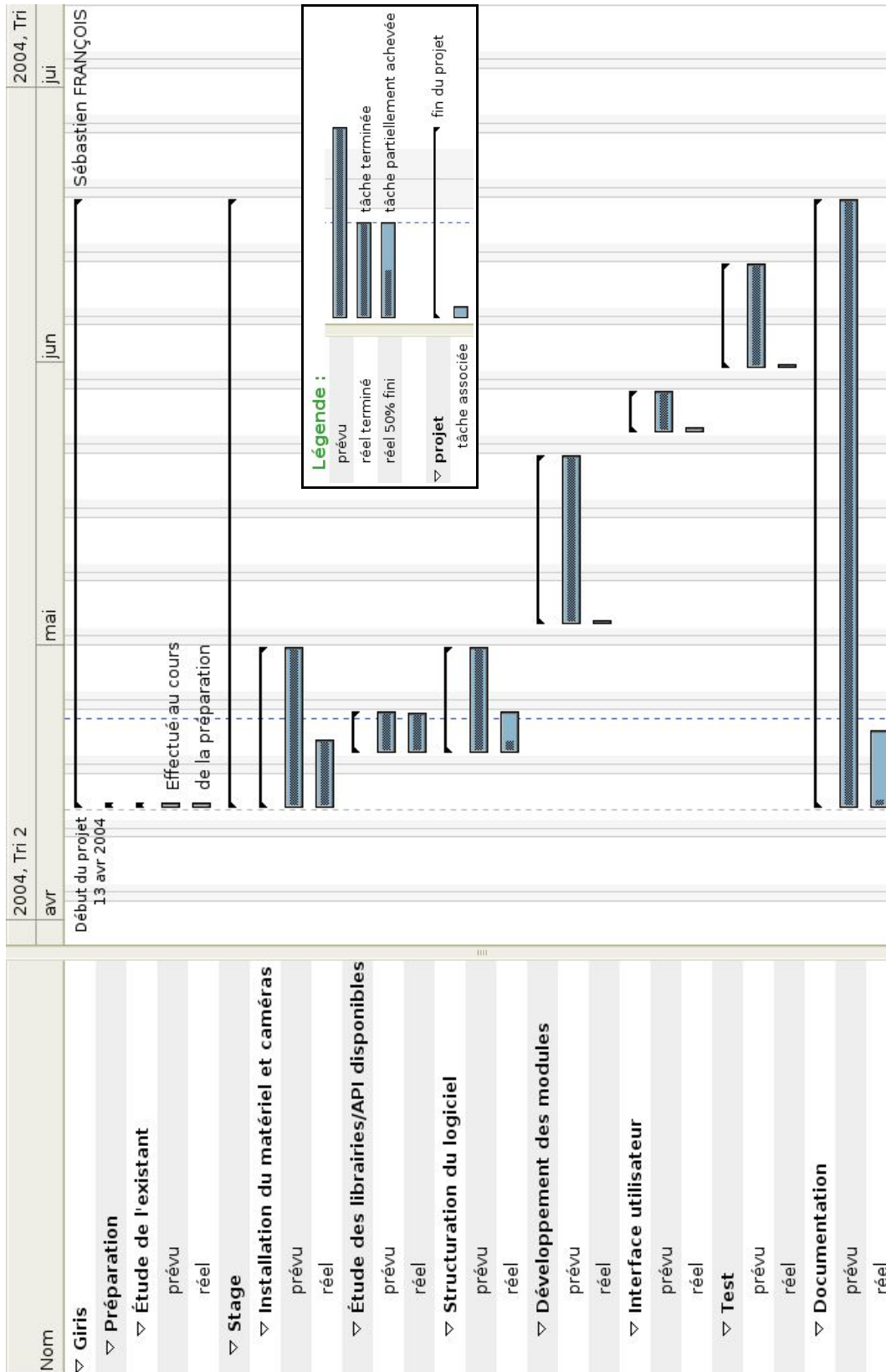
### 2.1. Méthodes et ressources

#### 2.1.1. Examen du cahier des charges

Ce projet nécessitera donc de faire appel au minimum à trois types de bibliothèques externes, pour l'acquisition, l'affichage mais aussi l'interface graphique.

L'acquisition devra supporter le plus de matériels possibles, l'affichage devra pouvoir exploiter au mieux les capacités du système et l'interface devra être agréable et simple à utiliser.

## 2.2. Planning prévisionnel



Dessin 1 : Planning prévisionnel (au 23 Avril 2004)

## 3. La réalisation

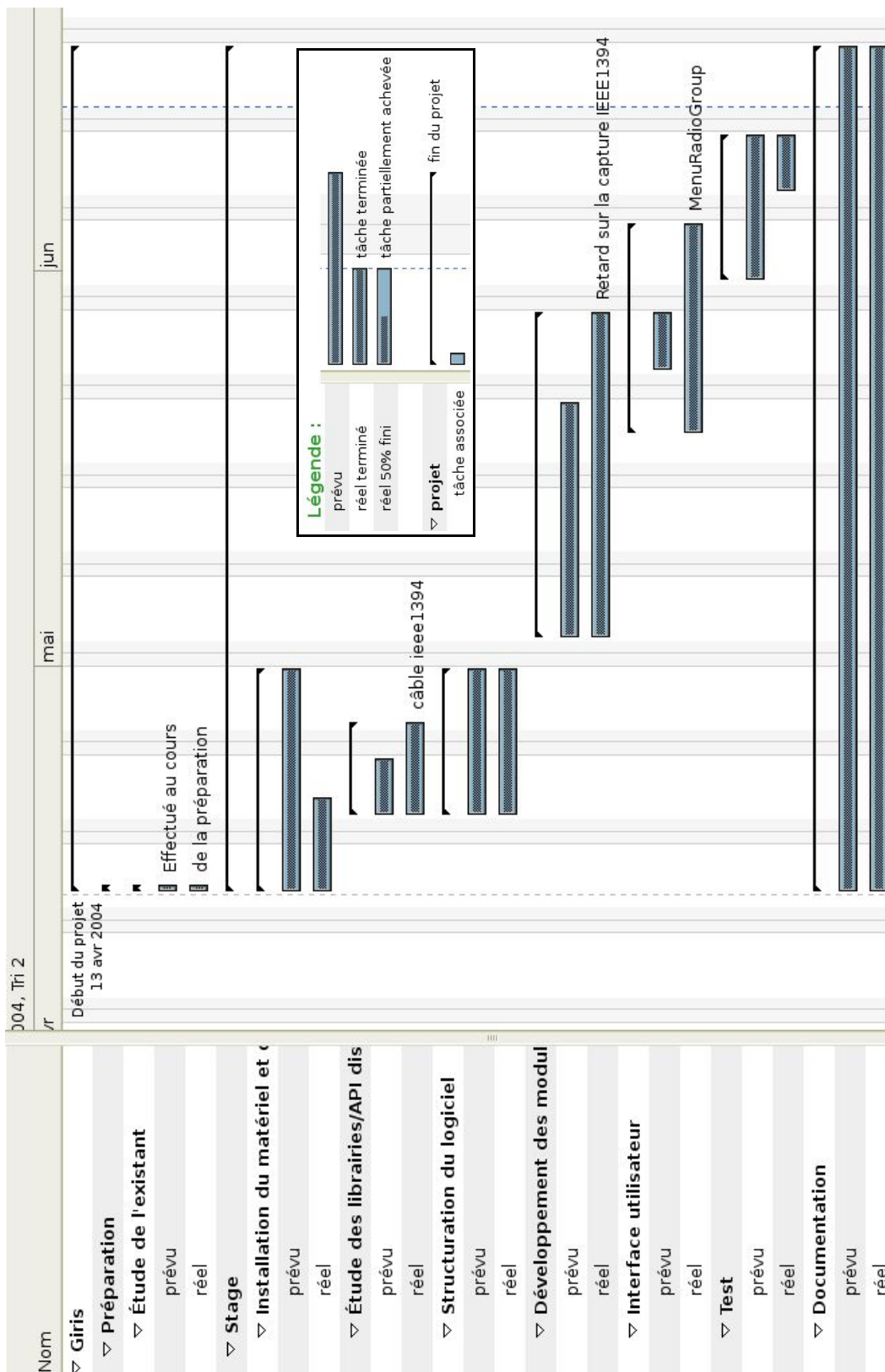
### 3.1. Présentation générale

Il n'y a pas eu d'évolution majeure du cahier des charges.

J'ai effectué une modélisation en **UML\*** à partir des cas d'utilisation que j'avais dégagés.

Nous avons convenu d'utiliser le langage C++, ce qui a orienté le développement vers une approche objet. M. SURUT m'a demandé de me consacrer à un fonctionnement simple de type capture-affichage, c'est à dire sans traitement explicite.

### 3.1.1. Planning réel



Dessin 2 : Planning réel (au 14 juin 2004)

## 3.2. Présentation détaillée des méthodes de travail

### 3.2.1. Installation de l'environnement

La première des tâches a été de m'assurer que la caméra USB fonctionne sous Linux.

Il m'a fallu utiliser un projet trouvé sur Internet pour l'utiliser dans une résolution acceptable, j'ai découvert un site fournissant un module et une librairie de décompression.

Sur le même site se trouve un logiciel, Camstream qui permet notamment d'utiliser les fonctions spécifiques de la ToUCam Pro II telles que la réduction de bruit, la qualité de la compression et les différents modes de fonctionnement, il m'a permis d'effectuer quelques tests.

Durant ces tests, j'ai observé un bogue dans l'ordre des octets de l'image, vraisemblablement une inversion des composantes Rouge et Verte.



*Camstream sur PowerPC*

*Camstream sur PC*

*XawTV sur PowerPC*

J'ai résolu ce problème avec le développeur Debian qui s'en occupe.

J'ai installé un poste sous Microsoft Windows 2000 et GNU\*/Linux Debian, pour avoir la possibilité d'observer le logiciel originel, d'utiliser Objectteering, mais aussi de pouvoir tester mon logiciel sur architecture PC.

Le support camescopie **DV\*** a été plus complexe : j'ai découvert en utilisant plusieurs logiciels (coriander, kino et dvgrab) que l'interface vidéo de haut niveau (video1394), ne le supportait pas (noyau 2.6.3). Il m'a fallu utiliser une autre interface dite raw1394, qui fonctionnait correctement, bien qu'étant de très bas niveau.

### 3.2.2. Choix des outils de développement

J'ai effectué différentes études sur les outils et les bibliothèques à utiliser, pour en arriver au choix d'associer **GTK\*** avec **SDL\***, **video4linux\*** et l'ensemble des bibliothèques gérant l'**IEEE1394\***.

M. SURUT m'a recommandé de tester OpenML. Lors des tests, tout ce qui concernait la vidéo ne fonctionnait pas, manquant vraisemblablement de maturité, ce n'était donc pas utilisable dans le cadre de ce projet. A l'avenir, il se pourrait que l'utilisation d'OpenML devienne beaucoup plus intéressante qu'un accès direct aux périphériques vidéos.

J'ai réalisé de petits programmes simples en C, me permettant de mieux comprendre le fonctionnement de l'**API\*** video4linux.

Je me suis renseigné en détail sur le fonctionnement de la vidéo sous BSD pour découvrir qu'en dehors de quelques projets isolés (BKTR), elle est quasiment inexistante, l'interface graphique elle-même n'est pas présente dans tous les « ports », j'ai donc abandonné l'idée de trouver un compromis.

J'ai ensuite étudié l'affichage, en examinant les différentes bibliothèques disponibles, j'ai mis au point un petit test comparatif entre l'utilisation d'OpenGL et celle de **SDL\***, l'examen des résultats a été sans équivoque et j'ai choisi **SDL\***.

En ce qui concerne l'interface graphique, je souhaitais avoir un document de référence concernant **Gnome\*** 2.0, car l'interface est assez récente, mais peu documentée. J'ai alors fait venir de l'étranger un livre très récent traitant du sujet :

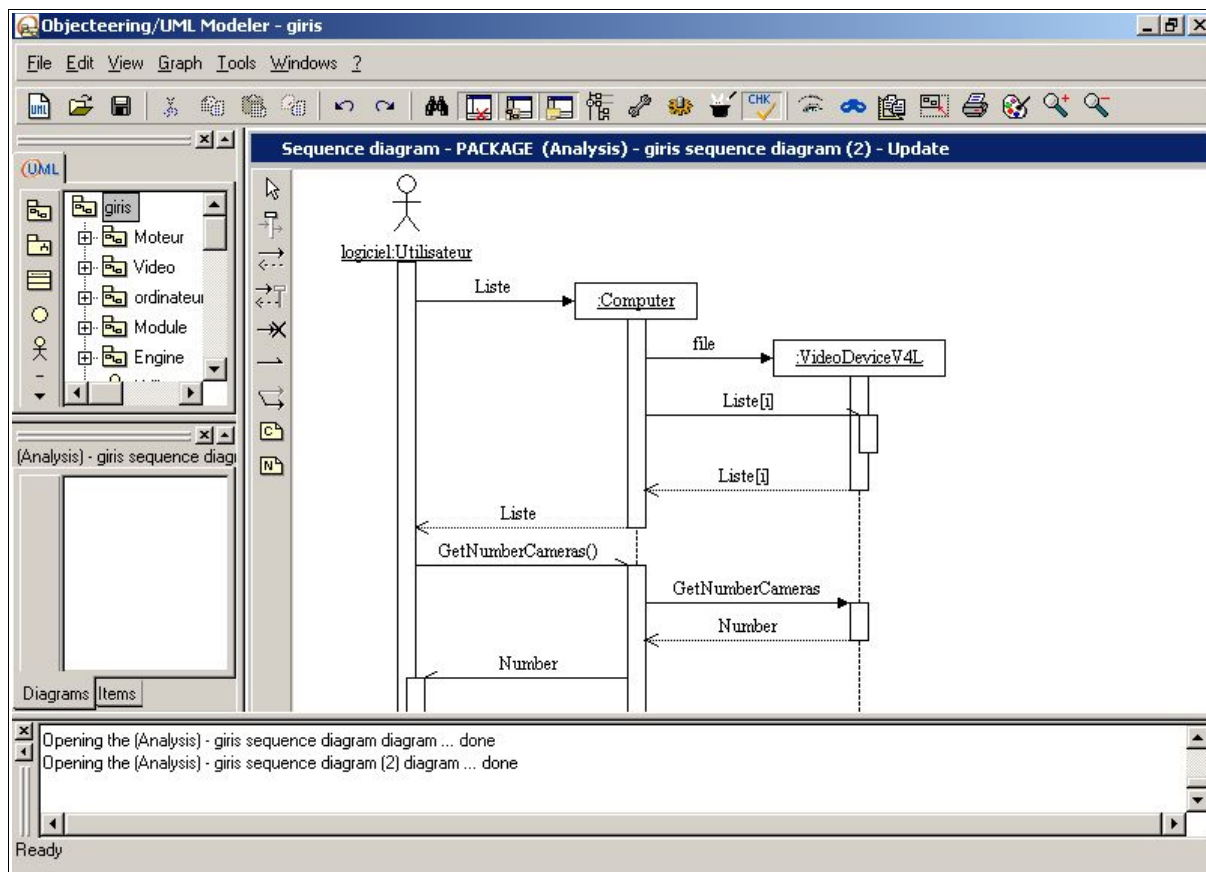
The Official Gnome 2.0 Developer's guide de Matthias Warkus.

Le 27 avril, M. SURUT a exprimé son souhait que le logiciel soit développé en C++, les bibliothèques de **Gnome\*** disposant d'une interface, les autres éléments devant être encapsulés.

### 3.2.3. La modélisation UML

La modélisation en **UML\*** m'a permis de me poser beaucoup de questions sur la structure à employer. J'ai commencé par rédiger un scénario simple, puis de ce scénario, j'ai tiré des cas d'utilisation que j'ai décrits selon un formulaire que j'ai traduit à partir d'un modèle fourni par M. SURUT.

Ensuite, j'ai réalisé un premier modèle, des différents objets que j'avais dégagé des scénarios, en utilisant le logiciel Objecteering, conformément aux habitudes de développement du service.



Aperçu 2 : Objecteering

Volontairement, la description est très générale, de manière à avoir un aperçu global de la tâche à accomplir et aussi de limiter la durée de cette modélisation à une semaine.

### 3.2.4. Portabilité

J'ai choisi d'intégrer à mon projet un ensemble d'outils (autoconf, automake et libtool) dont le but est de détecter les éléments requis pour compiler le projet et de suggérer l'installation des éléments manquants. Ils permettent aussi d'adapter les Makefiles aux systèmes sur lequel on construit le logiciel. Ils sont couramment employés dans le développement sous UNIX.

### 3.2.5. Structuration du logiciel / Documentation

J'ai développé les classes correspondant aux modèles et en essayant de garder des noms et une hiérarchie assez intuitifs.

J'ai mis en place un système reposant sur Doxygen, qui permet de générer la documentation de référence d'un programme à partir de commentaires placés dans le code. Je l'ai combiné à Graphviz pour obtenir en plus une représentation **UML\***.

J'ai défini un style de programmation pour essayer de garder une cohérence dans les sources.

### 3.2.6. Mise au point

Durant le développement, j'ai voulu tester les modules un par un, et il m'a fallu trouver comment accéder à l'information et vérifier sa pertinence.

J'ai surtout cherché à récupérer les images capturées et à pouvoir juger si le contenu était correct, il m'a fallu trouver un moyen de visualiser des images brutes aux formats **YUV\*** (YV12), **DV\*** et **RGB\***.

## 4. Le Bilan

### 4.1. Résultats obtenus

<i>Composant</i>	<i>État</i>
Capture video4linux1	fonctionne correctement
Capture IEEE1394	fonctionne en partie
Affichage SDL	fonctionne correctement
Interface graphique	fonctionne en partie

Lorsque l'on utilise une caméra et un caméscope, si l'on change plusieurs fois la caméra utilisée, un blocage peut apparaître (au niveau de la capture **IEEE1394\***).

Les drivers des deux caméras sont respectivement en Beta-1 et en expérimental, on peut donc difficilement déterminer la stabilité d'un logiciel les utilisant.

La sélection du format d'image n'est pas réglable, je n'ai pas développé les fonctions Config() des modules, de même pour le **moteur\***, dont on ne peut configurer les modules par l'interface graphique.



L'interface a un petit bogue, qui me force à sélectionner explicitement un élément du groupe de RadioMenu, sinon il y a un décalage entre le menu sélectionné graphique et celui dont la variable active est à 1, DanielK (auteur de l'interface C++ de **GTK\***) m'a confié que les RadioMenuItems n'étaient que très rarement utilisés et qu'il s'agissait bien d'un problème dans la librairie.

Il serait plus intéressant d'utiliser des listes chaînées pour la chaîne des modules.

Il serait souhaitable de développer un module d'acquisition V4L2.

## 4.2. Bilan de l'ensemble du projet

Je suis globalement satisfait du résultat, au-delà même du projet, la quantité de documentation consultée m'a permis d'apprendre beaucoup de choses sur le fonctionnement des périphériques vidéo, du noyau d'un système d'exploitation, du bus **IEEE1394\*** etc...

La principale difficulté a été de comprendre comment récupérer une image à partir d'un périphérique **IEEE1394\***, car ma seule référence, le code de dvgrab était très lourd et peu commenté.

Ma part de travail a été de structurer et de combiner un grand nombre de librairies et d'**API\*** existantes de bas niveau pour obtenir l'ébauche d'un logiciel simple d'utilisation, à l'interface agréable.

J'ai passé beaucoup de temps à lire des documentations et du code lorsqu'elles n'existaient pas. La conception structurelle a été assez courte, j'avais déjà une idée assez claire de ce que je souhaitais faire d'après mon travail de préparation.

## Glossaire

API	Application Programming Interface, interface utilisée par des applications pour accéder à une ressource.
Canal	Certains périphériques vidéo présentent plusieurs entrées, le canal correspond donc à l'entrée sélectionnée.
DV	Digital Video, format vidéo des caméscopes numériques.
FireWire	Nom donné par Apple aux périphériques IEEE1394.
Gnome	Projet visant à fournir un environnement du Bureau libre, regroupe un grand nombre d'outils d'aide au développement, dont <b>GTK*</b> .
GTK+	Boîte à outils, multiplateforme, pour créer des interfaces graphiques.
GNU	Acronyme de « GNU is Not Unix », projet dont le but est de fournir un système d'exploitation complet et libre et semblable à UNIX.
IEEE1394	Norme concernant les périphériques sur bus série haut débit, désigne donc ici les caméras numériques <b>DV*</b> .
Moteur	Bloc comportant l'ensemble des tâches liées à l'acquisition, au traitement, puis à l'affichage de l'image. On ne le contrôle qu'au travers de l'interface graphique.
Overlay	Peut désigner plusieurs types de transferts de données à la carte vidéo, ici il désigne une zone de l'affichage dans laquelle la carte graphique superpose directement des données brutes dans un format <b>YUV*</b> .
Paquet	« Un paquet est un logiciel qui forme une unité atomique. Un paquet contient en général un programme exécutable, les fichiers de données dont il a besoin et des catalogues de messages pour internationalisation et la documentation. » <a href="http://fink.sourceforge.net/doc/packaging/intro.php">http://fink.sourceforge.net/doc/packaging/intro.php</a>
RGB	Ensemble de formats vidéo dans lesquels les composantes sont le Rouge, le Vert et le Bleu (cf dossier technique Étude des formats d'images).
Scène	Environnement correspondant à l'image perçue par la caméra, l'image est le résultat de la conversion de la scène par la caméra. Elle peut par exemple comporter des problèmes d'éclairage partiel ou extrême.
SDL	Librairie gérant l'affichage accéléré, elle est utilisée pour les jeux vidéo (cf dossier technique Étude des bibliothèques disponibles).
Traitement	Bloc fonctionnel numérique auquel on fournit une image sous forme numérique et qui donne en sortie une image, il est possible de chaîner plusieurs de ces traitements.
UML	Unified Modeling Language, langage standard de modélisation.
V4L	Interface du noyau Linux aux webcams, cartes d'acquisition, et cartes TV.
video4linux	
Widget	Nom générique des éléments d'interface graphique tels que : bouton, une zone de saisie etc...
YUV	Ensemble de formats vidéo dans lesquels les composantes sont la luminance et deux informations de chrominance (cf dossier technique Étude des formats d'images).

# Bibliographie

## Interfaces graphiques

### Qt

[www.trolltech.com/products/qt](http://www.trolltech.com/products/qt)

[docs.kde.org/en/HEAD/kdevelop/kdearch/graphics.html](http://docs.kde.org/en/HEAD/kdevelop/kdearch/graphics.html)

Accélération matérielle des QPixmap

[doc.trolltech.com/3.3/qglwidget.html](http://doc.trolltech.com/3.3/qglwidget.html)

Documentation détaillée de l'objet QGLWidget

[doc.trolltech.com/3.3/qpixmap.html](http://doc.trolltech.com/3.3/qpixmap.html)

Documentation détaillée de l'objet QPixmap.

### GTK+

[www.gtk.org](http://www.gtk.org)

[mail.gnome.org/archives/gtk-list/2001-June/msg00233.html](mailto:mail.gnome.org/archives/gtk-list/2001-June/msg00233.html)

Discussion sur l'intégration de SDL dans **GTK\***

[www.libsdl.org/projects/gtk-demo](http://www.libsdl.org/projects/gtk-demo)

Intégration de **SDL\*** dans **GTK\* 1.2**

[gtkmm.org](http://gtkmm.org)

Site de l'interface C++ de **Gnome\***.

## Librairie d'affichage

[www.tldp.org/HOWTO/Linux-Gamers-HOWTO/x126.html](http://www.tldp.org/HOWTO/Linux-Gamers-HOWTO/x126.html)

**Peter Jay Salzman, Frédéric Delanoy**

Introduction aux différentes bibliothèques dans le HOWTO dédié aux jeux.

### SDL

[www.libsdl.org](http://www.libsdl.org)

[archives.seul.org/linuxgames/feb-2002/threads.html#00031](http://archives.seul.org/linuxgames/feb-2002/threads.html#00031)

Liste de diffusion linuxgames, débat OpenGL vs. **SDL\*** pour jeux en 2D.

### Clanlib

[www.clanlib.org](http://www.clanlib.org)

[www.happypenguin.org/forums/viewtopic.php?t=1273](http://www.happypenguin.org/forums/viewtopic.php?t=1273)

Forum de développement de jeux, qu'est-ce que Clanlib par rapport à **SDL\***

[www.clanlib.org/intro.html](http://www.clanlib.org/intro.html)

Introduction, où l'on trouve la phrase mentionnant les problèmes de portabilité.

### OpenML

[www.khronos.org/openml/](http://www.khronos.org/openml/)

### Video4linux

[www.exploits.org/v4l/](http://www.exploits.org/v4l/)

Annuaire de liens sur le sujet

Arborescence du noyau Linux

Spécification de l'interface de programmation vidéo pour Linux

[Documentation/video4linux/API.html](http://Documentation/video4linux/API.html)

[kernelnewbies.org/documents/kdoc/videobook/v4lguide.html](http://kernelnewbies.org/documents/kdoc/videobook/v4lguide.html)

Guide de mise au point d'un driver (module) pour périphérique vidéo

**Alan Cox**

[v4l2spec.bytesex.org/spec/](http://v4l2spec.bytesex.org/spec/)

**Bill Dirks, Michael H. Schimeki**

Spécification de la nouvelle interface de programmation vidéo pour Linux.

## BSD

[lists.freebsd.org/pipermail/freebsd-multimedia/2003-](http://lists.freebsd.org/pipermail/freebsd-multimedia/2003-July/thread.html#328)

[July/thread.html#328](http://lists.freebsd.org/pipermail/freebsd-multimedia/2003-July/thread.html#328)

[resin.csoft.net/cgi-bin/man.cgi?sektion=4&topic=bktr](http://resin.csoft.net/cgi-bin/man.cgi?sektion=4&topic=bktr)

Absence de compatibilité de BSD pour V4L

Manuel du programmeur BSD sur l'API vidéo.

## IEEE1394

[linux1394.org](http://linux1394.org)

Site officiel du support sous Linux.

## Formats d'images

### RGB\*

[en.wikipedia.org/wiki/RGB](http://en.wikipedia.org/wiki/RGB)

Article de la Wikipedia.

### YUV\*

[en.wikipedia.org/wiki/YUV](http://en.wikipedia.org/wiki/YUV)

[fourcc.org](http://fourcc.org)

Article de la Wikipedia.

Explications sur les différents formats YUV.

## Outils

[sources.redhat.com/autobook](http://sources.redhat.com/autobook)

**GNU\*** autoconf, automake and libtool.

## Ouvrages lus dans le cadre du stage

Language C

**Philippe DAX**

**Editions Eyrolles**

Ouvrage sur le développement C sous UNIX

Linux secrets

**Naba BARKAKATI**

**Editions SYBEX**

Aperçu global du système avec quelques rubriques consacrées au développement

Modélisation : objets avec UML

**Pierre-Alain Muller, Nathalie Gaertner**

**Editions Eyrolles**

Guide sur la modélisation d'objets en **UML\***

Programmation système C sous Linux

**Christophe BLAESS**

**Editions Eyrolles**

Présentation globale du développement sous **GNU\*/Linux**

Programmation Linux avec GTK+

**David ODIN**

**Editions Eyrolles**

Guide à la programmation de GTK 1.2 en C

The official Gnome 2.0 developer's guide

**Matthias Warkus**

**Editions No Starch Press**

Guide de référence pour **Gnome\*** 2.0.

## Dimension professionnelle du stage et vécu

### 1. Dimension professionnelle

#### 1.1. Historique

Le 10 juin 1885, Mademoiselle MULOT, Directrice d'école privée, fonda la première école départementale pour jeunes déficients visuels, où la méthode Braille était enseignée.

En septembre 1926, l'école fut transférée aux hospices d'Angers près du chemin de Montéclair, au sein d'un quartier résidentiel et à quelques minutes du centre ville.

Depuis 1985, de nouveaux locaux ont été aménagés par l'Institut, au Lac de Maine à Angers.

Le 1er Janvier 1992, une mini-crèche a été créée, permettant de rendre l'Institut moins spécifique aux déficiences sensorielles.

Aujourd'hui, l'Institut Montéclair est un centre mutualiste de rééducation pour déficients visuels, il est géré par la Mutualité de l'Anjou.

Il accueille les jeunes pour des formations, ou bien pour l'internat, leurs études à proprement parler sont effectuées au sein d'établissements scolaires habituels, dans un souci d'intégration. L'Institut fournit des services adaptés à tous les âges. L'insertion professionnelle est aussi très importante pour leur assurer une autonomie, l'Institut dispose par exemple d'ergonomes dont la mission est l'adaptation des postes de travail.

Au sein de l'Institut sont aussi présents des services pour malentendants, rattachés à l'Institut Charlotte Blouin d'Angers.

#### 1.2. Domaines de compétences de l'Institut

##### Les jeunes

L'Institut intervient pour assurer le suivi et l'intégration de jeunes déficients visuels allant de la grande section maternelle à l'enseignement supérieur, en collaboration avec des établissements angevins. Ils sont intégrés au sein de classes spécialisées ou bien en classes ordinaires. L'Institut les forme individuellement à acquérir des techniques palliatives dans le domaine de la communication écrite et à savoir manipuler des aides techniques telles que terminaux brailles, outils de bureautique, etc...

Les élèves, résidant loin d'Angers, peuvent être internes, l'Institut dispose de pavillons où des éducateurs spécialisés les aident à s'organiser au quotidien et assurent leur suivi scolaire. L'Institut met à la disposition des jeunes du matériel et des techniques spécifiques pour pallier à leur handicap et leur permettre de pouvoir suivre leur scolarité dans les meilleures conditions.

## **Les adultes**

Le Service Interrégional d'Appui aux Adultes Déficients Visuels a pour but d'aider une personne dans le cadre professionnel, au travers de différents aspects :

1. Informer / sensibiliser sur les capacités professionnelles des personnes déficientes visuelles, présenter les techniques palliatives et les formations et métiers accessibles
2. Réaliser un bilan fonctionnel pour permettre à une personne d'évaluer sa vue, son autonomie dans les déplacements et dans la vie quotidienne, ses compétences en bureautique et informatique
3. Appuyer une personne durant dans le cadre de formations professionnelles ou universitaires, par une adaptation des documents mais aussi des outils de l'environnement, notamment en informatique
4. Accompagner techniquement une personne à l'insertion ou au maintien dans l'emploi, par l'intervention d'ergonomes, de prêts de matériels, etc...

## **Centre de formations**

L'Institut Montéclair est agréé pour la formation continue, il propose des stages de formations spécifiques à la basse vision et destinés aux professionnels concernés (opticiens, personnel de maison de retraite...).

## **Centre de recherche informatique**

L'Institut a allié ses connaissances de la basse vision et les techniques de l'informatique pour produire des logiciels de rééducation adaptés aux malvoyants.

### 1.3. Service recherche et développement

Ce service a été créé pour développer des logiciels d'évaluation et de rééducation pour la basse vision. En octobre 1998, une EURL a été fondée pour commercialiser ces produits.

L'équipe fait collaborer orthoptistes et informaticiens dans le but de créer des logiciels spécifiques répondant aux besoins exprimés sur le terrain.

A ce jour, il existe les logiciels suivants :

- évaluation de la vision de loin
- évaluation de la vision de près
- aide à la lecture
- rééducation motrice
- travail de la mémoire visuelle.

Plusieurs prototypes et concepts sont aussi développés :

- poste de travail bi-écran permettant de séparer l'espace de travail du patient de celui de l'orthoptiste
- rééducation à domicile assistée par l'orthoptiste.

### 1.4. Contacts avec les autres services

J'ai beaucoup apprécié la visite de l'Institut, qui a permis de mieux voir la structure au sein de laquelle je travaillais, et de réaliser l'étendue des domaines que couvrent les services de l'Institut Montclair. Ce fut aussi l'occasion de découvrir les différents métiers auxquels l'Institut fait appel.

J'ai eu l'occasion de pouvoir partager mes connaissances de l'administration de serveur de type UNIX et d'utiliser mes relations au sein du fournisseur d'accès Internet Free pour obtenir de l'aide.

J'ai été heureux de pouvoir découvrir le travail des informaticiens au sein d'un service où régnait une bonne entente.

J'ai beaucoup échangé avec des personnes responsables de services liés à l'aide aux déficients auditifs, car si je connaissais le monde des déficients visuels, celui des sourds m'était complètement inconnu.

## 1.5. Contact avec l'extérieur

Le 9 juin, je me suis rendu avec M. SURUT, à Thalès Recherche et Technologie, situé à Orsay. Ce fut l'occasion de rencontrer Joseph COLINEAU, auteur de PortaNum, et Loïc GOASDUFF, responsable de la communication autour de PortaNum. Le but de cette rencontre était de suivre un cours de traitement d'image.

Au cours d'une réunion, nous nous sommes exprimés à tour de rôle à propos du projet, ce fut pour moi l'occasion de leur présenter mon travail et de pouvoir expliquer quelles nouvelles applications pouvaient en découler. Joseph COLINEAU a exprimé le souhait de rendre les traitements de PortaNum disponibles sous la forme d'une librairie, de manière à pouvoir intégrer facilement les traitements à d'autres applications comme Giris.

## 2. Vécu du stage

Ce stage m'a permis d'acquérir des connaissances dans des domaines très différents : de la programmation matérielle de bas niveau, à celle orientée objet, en passant par différents formats d'images, et ce sous un environnement fondamentalement différent.

J'ai eu l'occasion de voir de plus près les manques et défauts du développement libre, la contrainte de travailler avec une documentation sommaire, si existante. Il m'a donc fallu progressivement intégrer des bribes d'informations récupérées dans les fichiers d'en-tête, des sites, de discussions sur IRC.

J'ai beaucoup apprécié cependant d'avoir pu avoir cette expérience contrastée, qui m'a permis de mieux savoir ce qu'était le développement sous un UNIX tel que GNU/Linux.

Ce stage m'a conforté dans l'idée de poursuivre mes études dans le domaine informatique tout en laissant une part prépondérante à l'aspect social. J'ai regretté le manque de contacts avec les autres services qu'a impliqué le travail que j'ai effectué. Après avoir vu une interprète signer pour une personne atteinte de surdité, j'ai décidé d'apprendre la langue des signes.



### 3. Mentions légales

**PortaNum** est une marque commerciale de THALÈS.

**Windows** est une marque déposée de Microsoft Corporation.

**Linux** est une marque déposée de Linus Torvalds.

**Mac** est une marque déposée d'Apple Computer, Inc.

**Unix** est une marque déposée de The Open Group.

**BSD** est une marque déposée de Berkeley Software Design, Inc.

**OpenML et OpenGL** sont des marques déposées de Silicon Graphics, Inc.

Toutes les autres marques sont détenues par leur propriétaire respectif.

## 4.a. Résumé du projet

Ce document montre quelle a été ma démarche pour mener à bien l'étude et la réalisation du projet qui m'a été confié.

Ce projet a pour but l'étude de la transposition de PortaNum, logiciel de la société Thalès, existant sous Microsoft Windows, vers un autre système d'exploitation : GNU/Linux.

Le logiciel originel permet la vision de documents (vision de près) ou bien d'éléments distants (vision de loin) en adaptant l'image aux déficiences de l'utilisateur.

Le développement dans le cadre de ce stage était limité à la capture et l'affichage d'une image provenant de deux périphériques vidéo : une webcam USB et un caméscope numérique Firewire.

Les modules de traitement du logiciel originel devront être adaptés à la structure modulaire de Giris, ensuite le logiciel pourra être utilisé comme PortaNum.

## 4.b. Project Abstract

This document shows the different steps of the study and the realization of the project I was given.

This project aims at porting PortaNum, a Thalès software, which currently exists for Microsoft Windows, to another operating system: GNU/Linux.

The original software enables the vision of documents (near sight) or of distant elements, like a whiteboard (far sight) by adapting the picture according to the user's deficiencies.

The development in the context of this internship was limited to the capture and display of a picture taken from two kinds of video devices : a USB webcam and FireWire digital video camera.

The processing modules used in the original software will need to be adapted to the modular structure of Giris, then the software can be used like PortaNum.

