N°d'ordre :

N° de série :

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED

FACULTÉ DES SCIENCES EXACTES

Département D'Informatique

Mémoire de Fin D'étude

Présenté pour l'obtention du Diplôme de

LICENCE ACADEMIQUE

Domaine : **Mathématique et Informatique**

Filière : **Informatique**

Spécialité : **Systèmes Informatiques**

Thème

Développement d'un outil de design des diagrammes UML pour LaTeX

Présenté par :

- **BENAMOR Kamal**
- **NAAMI Aissa**

Proposé et Encadré par : **M. ZAIZ Faouzi**

Soutenue le 06-06- 2018 Devant le jury:

M. ABBAS Messaoud Président

Mme. GUIA Sana Sahar Rapporteur

Année Universitaire: 2017-2018

Remerciements

Nous remercions le bon Dieu, tout puissant, de nous avoir donné la force pour suivre, ainsi que l'audace pour dépasser toutes les difficultés.

Nous souhaitions adresser nos Remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

On tient à remercier sincèrement Mr. ZAIZ Faouzi, qui a toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire.

Les jurys pour leurs efforts et leur soin apporté à notre travail.

Aux enseignants de notre département informatique de l'université d'echahid Hamma Lakhder d'El Oued.

Enfin, nous adressons nos plus sincères remerciement à tous nos proches et amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Dédicaces

Je dédie ce modeste travail à :

Mon père et ma mère ; vous qui étaient toujours à mes côtés pour me soutenir et m'encourager à me battre sans jamais m'arrêter à mi-chemin; que dieu vous protège.

A Mes Frères et sœurs et Ma femme.

Toute la famille Benamor.

Tous Ma chère frère : Mounir

A mes collègues de travail de
l'agence locale de l'emploi de djamâa

A Mon binôme: NAAMI Aissa.

A la promotion 3éme LMD INFORMATIQUE 2017/2018

Université Echahid Hamma Lakhdar d'El Oued.

Kamal

Dédicaces

Je dédie ce modeste travail à:

Mon Père et Ma mère pour leurs soutiens, leurs amours, et ses encouragements Que Dieu les garde.

A Mes Frères et sœurs.

Toute la famille Naami.

A Mon binôme: BENAMOR Kamal.

A la promotion 3éme LMD INFORMATIQUE 2017/2018

Université Echahid Hamma Lakhdar d'El Oued.

Aissa

Résumé

TeX a été développé par Donald Knuth en 1978 pour aider les scientifiques à publier des articles. LaTeX est écrit par Leslie LAMPORT en 1982, il est capable de produire des différents documents comme les articles, livres, rapports, lettres, diagrammes, …etc. LaTex est utilisé par les étudiants et les chercheurs dans le monde universitaire et scientifique telque dans les domaines Informatique, Mathématique et Physique pour crée du document de qualité élevée.

Dans ce travail Nous avons développé un outil de design des diagrammes (Formes géométriques simples par exemple : ligne, flèche , triangle carré, rectangle, rectangle arrondi, , cercle, ellipse, , losange) qui facilité la tâche de design (ajouter une forme, insérer un texte dans la forme, colorer le fond de la forme ou le contour, agrandir ou réduire une forme, déplacer une forme, supprimer une forme, Attachement des formes en utilisant une ligne ou une flèche , sauvegarder un diagramme...), accélère le rôle de l'utilisateur est simplifie la génération automatique du code des diagrammes (le code produit concerne la bibliothèque TikZ) à travers une interface graphique.

Mots Clés : LaTeX, PostScript, Diagramme, TikZ.

Abstract

TeX was developed by Donald Knuth in 1978 to help scientists publish articles. LaTeX is written by Leslie LAMPORT in 1982, able to produce different documents like articles, books, reports, letters, diagrammes …etc. LaTex is used by students and researchers in the academic and scientific world such as Computer Science, Mathematics and Physics for high quality document products.

In this work we have developed a tool for the design of diagrams (simple geometric shapes for example: line, arrow, square triangle, rectangle, rounded rectangle, circle, ellipse, rhombus) which facilitates the task of design (add a shape, insert a text in the shape, color the bottom of the shape or the outline, enlarge or reduce a shape, move a shape, delete a shape, Attach shapes using a line or an arrow, save a diagram ...), accelerates the role of the user is simplifies the automatic generation of code diagrams (the product code relates to the library TikZ) through a graphical interface.

Key words: LaTeX, PostScript, Diagram, TikZ.

ملخص

تم تطوير TeX من قبل دونالد كنوث في عام 1978 لمساعدة العلماء على نشر المقالات. كتب LaTeX من قبل ليزلي LAMPORT في عام 1982، قادرة على إنتاج وثائق مختلفة مثل المقالات والكتب والتقارير والخطابات والرسوم البيانية، ... الخ

يتم استخدام LaTex من قبل الطلاب والباحثين في العالم الأكاديمي والعلمي مثل علوم الكمبيوتر والرياضيات والفيزياء لمنتجات المستندات عالية الجودة.

في هذا العمل قمنا بتطوير أداة لتصميم الرسوم البيانية (الأشكال الهندسية البسيطة على سبيل المثال: الخط ، السهم ، المثلث المربع ، المستطيل ، المستطيل الدائري ، الدائرة ، القطع الناقص ، المعين) الذي يسهل مهمة التصميم (إضافة شكل ، إدراج نص في الشكل ، أو لون الجزء السفلي من الشكل أو المخطط التفصيلي ، أو تكبير أو تصغير شكل ، أو تحريك شكل ، أو حذف شكل ، أو إرفاق أشكال باستخدام خط أو سهم ، أو حفظ رسم بياني ...) ، تسريع دور المستخدم يبسط الجيل التلقائي من الرسوم البيانية رمز (رمز المنتج يتصل مكتبة TikZ) من خلال واجهة رسومية.

الكلمات المفتاحية:LaTeX، PostScript، Diagram، TikZ

Table des matières

Liste des figures

Introduction générale

TeX est un logiciel de mise en page conçu par Donald Ervin KNUTH à la fin des années 70. LaTeX est écrit par Leslie LAMPORT en 1982, est un jeu de macros par dessus TeX, plus facile à utiliser. LaTeX a été conçu pour rédiger des articles, des rapports, des thèses, des livres ou pour préparer des transparents. On peut insérer dans le texte, des dessins, des tableaux, des formules mathématiques et des images sans se soucier des détails de la mise en page. Les documents produits avec LaTeX et TeX sont d'une excellente qualité typographique.

Il est très difficile de générer manuellement le code de certains éléments tels que les tableaux ou les diagrammes. Dans ce travail nous allons proposer un système qui permet de générer le code d'un diagramme LaTeX en TikZ avec une erreur d'affichage graphique minimale. Afin de réaliser cette tâche le système doit transformer les différentes formes affichées dans l'interface en utilisant le langage C# en une séquence de code de formes en TikZ. Cette transformation génère les problèmes suivants :

- C# utilise un système d'affichage en coordonnées d'écran autant que TikZ utilise un système d'affichage en coordonnées cartésien.
- C# utilise un système de positionnement des formes à base de coin ou point supérieur gauche autant que TikZ utilise un système de positionnement des formes à base du centre.
- Certains éléments sont dessinés en C# différemment qu'en TikZ, tel qu'un rectangle arrondi.

Le reste du travail est organisé comme suit :

Le premier chapitre présente un état de l'art sur LaTeX et différents systèmes de graphisme utilisé.

Le second chapitre contient la conception et la mise en œuvre du système à développer.

Le troisième chapitre présente le choix du langage de programmation ainsi qu'une petite présentation de l'utilisation de l'interface de l'application.

Nous terminons le travail par une conclusion sur les résultats obtenus par les méthodes utilisées, et des perspectives de ce travail.

Chapitre 1 : Etat de l'Art

Introduction

LaTeX représente aujourd'hui le langage d'écriture de toute la communauté mathématique et scientifique. Contenant votre texte et quelques commandes de mise en page. Ce code sera traité pour finalement résulter un document prêt à être imprimé.

Dans ce chapitre, nous allons voir un état de l'art sur LaTeX ainsi qu'un petit survol sur les différents systèmes d'affichage graphique tel que TikZ et Pstricks.

1.1 Bref Historique :

1978 : Donald E. Knuth désire créer un programme permettant d'obtenir des documents de qualité typographique irréprochable. Il crée alors le formateur de texte TeX, avec deux objectifs :

Qualité : permettre de produire les meilleurs documents

Pérennité : être aussi indépendant que possible des mutations technologiques affin que les archives constituées par les documents TeX restent utilisables très longtemps.

En 1982 : Leslie Lamport crée *LaTeX*, une autre couche de macros sur TeX avec le même objectif (simplifier l'emploi de TeX), avec la source au monde du logiciel libre

En 1985 : la livraison de la première version de LaTeX basée sur TeX comme outil de mise en page.

1994 : l'utilisation de la version stable LaTeX2ε tout en attendant LaTeX3

Fin des années 1990 : Hàn the thành introduit le moteur **pdftex**(son travail de thèse) :

Sortie par défaut dans le format **PDF** (portable document format conçu par Adobe), administration des polices vectorielles, extensions micro-typographiques, accès à des fonctionnalités PDF (hyperliens, table des matières. . .).

2008 : Le moteur **tex** touché la version 3.1415926 (corrections des erreurs). Aucune fonctionnalité n'est additionnée à **tex** depuis la version3. Pour chaque correction de faute ajoute une décimale de π.

en même année (2008) : La première version publique du moteur **xetex**.
Extension de **pdftex** pour utiliser les polices installées sur le système d'exploitation, codification en entrée Unicode (16 bits).
2010 : La première version publique du moteur **luatex**. Fusion du meilleur de **pdftex** et de **xetex**, ouverture de l'assemblage des pages au langage de programmation Lua.
2015 : **xetex** et **luatex** remplacer **pdftex** : utilisation des dernières technologies en domaine de polices vectorielles (TrueType, OpenType), en particulier le "standard" développé par Microsoft et Adobe sur les polices mathématiques.
Futur proche : Le projet LaTeX3 doit remplacer LaTeX2e. La commande LaTeX correspond en fait à lancer **tex** avec le format LaTeX (latex.fmt). [3]

1.2 Qu'est-ce que LaTeX ?

LaTeX est un logiciel de processeur de texte mais à la différence des autres, LaTeX demande au rédacteur de se concentrer sur la structure logique de son document, son contenu, tandis que la mise en page du document Contrairement aux logiciels de traitement de texte comme Microsoft Word où l'on voit immédiatement à l'écran le document tel qu'il sera imprimé, LaTeX lit un fichier source contenant à la fois le texte et des commandes de mise en page et génère un fichier portant le suffixe **.dvi** (pour DeVice Independent). De nos jours, ce fichier **dvi** est de moins en moins utilisé. En effet, LaTeX peut maintenant produire directement un fichier en format **PDF** facile à visualiser à l'écran ou à imprimer. [1]

La production d'un document LaTeX comporte généralement les trois étapes suivantes:

- écriture du fichier source
- compilation du fichier source
- visualisation et impression du document

1.3 Pourquoi utiliser LaTeX?

Le fichier source est du texte.
– Taille très petite : quelques Mo pour un livre de 600 pages.
– Très grande portabilité (tous les systèmes d'exploitation).

– Le texte peut être généré par un logiciel tiers (insertion dans un flux automatisé).
– Logiciels gratuits, ouverts et stabilisés

– pérennité des documents.
– Typographie de très grande qualité due au moteur tex : césures, ligatures...
– Possibilité de programmation : macros personnelles, mise en page, aspect des éléments du texte, automatisations diverses...
– Séparation du fond et de la forme.
– Changement de style aisé.
– Gestion automatique de nombreux éléments du document (table des matières, références croisées, bibliographie...).
– Capacité à gérer des gros documents complexes.
– Gestion aisée de documents écrits dans des langues et dialectes différents.
– très utilisé en linguistique.
– Nombreuses extensions sous forme de *packages*.
– Écosystème riche : makeindex, bibtex, metapost...
– Last but not least : excellente composition des formules mathématiques. [6]

1.4 Comment installer LaTeX ?

Il faut tout d'abord installer une distribution correspondant au système d'exploitation :
1) sous Unix : TeXLive
2) sous Mac : MacTEX
3) sous Windows : MiKTEX
À ceci s'ajoute l'installation d'un éditeur de texte. Par exemple emacs sous Unix, TeXMaker ou TeXnicCenter sous Windows, TeXshop ou iTeXmax sous mac.
À ceci se rajoute la possibilité d'installer une version portable sur une clé Usb permettant de travailler sur un poste non équipé.

Produire un document LaTeX consiste à traduire (on dit aussi compiler) une source créée par un éditeur de texte en un format destiné à l'affichage ou à l'impression. On notera (entre autres) les formats DVI, Postscript ou PDF. [5]

1.5 Les concepts de base de LaTeX :

Avant de pouvoir utiliser pleinement LaTeX et de profiter de sa puissance,il faut comprendre certains concepts.et vous initiera plus profondément à la philosophie et à

l'esprit de LaTeX.

1.5.1 Les fichiers LaTeX :

LaTeX est un langage de programmation, qui génère plusieurs types de fichiers. On trouve des fichiers:

.tex : Ce sont les fichiers contenant toutes les commandes que vous allez taper (fichiers sources)

.dvi : C'est le résultat de la compilation standard de vos commandes. On peut visualiser ces fichiers à l'aide du logiciel xdvi

.ps ou .pdf : Il s'agit des fichiers destinés à la publication, après conversion depuis le .dvi

.bib et .bbl : Ces fichiers servent à la gestion de la bibliographie.

.aux, .toc, .idx : Ces fichiers sont utilisés par LaTeX pour gérer les références dans votre document. [2]

1.5.2 Syntaxe d'une commande LaTeX :

On reconnaît ici la syntaxe générale d'une commande LaTeX :

\commande[*option*]{*argument*}

– une commande commence par une barre oblique inversée \;

– celle-ci est suivie par le nom de la commande;

– viennent ensuite, s'il y en a, les arguments optionnels, entre crochets,séparés par des virgules s'il y en a plusieurs;

– puis les arguments obligatoires, entre accolades, séparés par des virgules s'il y en a plusieurs. [4]

1.5.3 Le préambule :

Tout documenst LaTeX possède un préambule dans lequel figurent des informations valides pour l'ensemble du texte.

Le préambule débute à la première ligne du fichier et se termine à la *balise* **\begin{document} (exclus)**.

Tout ce qui est écrit entre \begin{document} et \end{document} constitue le corps du document.

Rien de ce qui est marqué ensuite n'est pris en compte. [4]

1.5. 4 La classe d'un document :

La classe du document définit sa structure physique.

Chaque classe a ses propres règles de mise en page et certaines commandes particulières.

Classes: article, letter, report, book, beamer . . .

Options: 11 pt, a4paper, landscape, twocolumn, oneside . . .

Certains caractères sont réservés: \{ } % # ~& $ ^_

On pourra obtenir les accents souhaités : à á â ä ã a (possibilité sur les majuscules) ainsi que des caractères spéciaux : æ œ ç

et les accents en mode mathématiques : $a \hat{} a \sim a \bar{} a$ *[6]*

1.5.5 Les packages (extensions ou modules) :

Une *extension* (*package* en anglais) sert à modifier la mise en pages ou à définir de nouvelles commandes. Ce sont les extensions qui permettent d'ajouter de nouvelles fonctions à LaTeX. Elles sont chargées grâce à la commande **\usepackage**, utilisée dans le préambule du document. [1]

1.5.6Références :

On utilise BibTeX afin d'insérer des références bibliographiques. Il faut tout d'abord créer un fichier comprenant les références. Ce fichier doit être écrit en format texte, avoir l'extension .bib et être constitué d'une succession de définitions de références.[1]

1.6 Qu'est-ce que PostScript ?.

C'est un Langage déterminé par la société Adobe pour reproduire un document destiné à l'impression. Ce langage assemblé de primitives de bas niveau peut être traduit par des programmes pour exécuter des aperçus avant impression ou directement par des circuits électroniques embarqués sur les imprimantes pour présenter l'image à imprimer. [8]

1.7 Qu'est-ce que PSTricks ?

PSTricks est une extension de LaTeX qui permet d'utiliser la majure partie des possibilités de PostScript pour le dessin à l'aide des commandes directement utilisables depuis LaTex.

1.8 Qu'est-ce que PGF ?

PGF (Portable Graphics Format) est une extension à LaTeX qui permet de générer des graphiques à l'aide de commandes en-ligne.

1.9 Qu'est-ce que TikZ?

Ti*k*Z est un package pour LaTeX permettant d'inclure des figures au format PDF en restant dans l'environnement LaTeX.

Il a été créé vers 2006 par Till Tantau. Il devient rapidement populaire, car il répond aux besoins précédents en évitant les inconvénients des autres solutions. La phase initiale d'apprentissage est rapide, et les figures simples peuvent être obtenues simplement. On sent que le langage a été conçu pour répondre à des besoins usuels de manière pratique.

Il continue d'évoluer, et les extensions actuelles permettent de créer des illustrations très variées.

Utiliser Ti*k*Z est un plaisir car on obtient des figures précises et d'une grande qualité, avec une impression de maîtrise. [8]

1.10 Quelques exemples pour design des différent formes sous LaTeX en utilise le package TikZ :

Dessiner une ligne :


```
\ begin { tikzpicture }
\draw (0 ,0) −− ( 0 , 5 ) ;
\end{ tikzpicture }
```

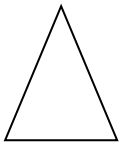
Dessiner une flèche :


```
\begin{tikzpicture}
\draw [->] (0,0) – (4,0) ;
\end{tikzpicture}
```

Dessiner une flèche cassée :


```
\begin{tikzpicture}
\draw [->,dashed] (0,0) – (6,0) ;
\end{tikzpicture}
```

Dessiner un triangle :



```
\begin{tikzpicture}
\draw (0,0) – (1,2)–(2,0)–cycle;
\end{tikzpicture}
```

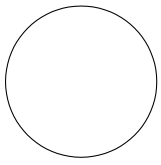
Dessiner un rectangle :

```
\begin{tikzpicture}
\draw (-1,-.2) rectangle ++(2,1);
\end{tikzpicture}
```

Dessiner un rectangle avec des angles ronds :

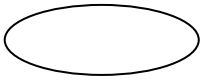
```
\begin{tikzpicture}
\draw [rounded corners](0,0) rectangle(2,1);
\end{tikzpicture}
```

Dessiner un cercle :



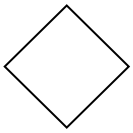
```
\begin{tikzpicture}
\draw (0,0) circle (1);
\end{tikzpicture}
```

Dessiner une ellipse :



```
\begin{tikzpicture}
\draw (1,1) ellipse (1 and .3);
\end{tikzpicture}
```

Dessiner un losange



```
\begin{tikzpicture}
\draw (1,0) – (0,1) – (-1,0) – (0,-1)– cycle ;
\end{tikzpicture}
```

Conclusion :

Dans ce premier chapitre nous avons vu quelques définitions de base sur LaTeX (document, commande, classe,…).En suite nous avons touché les systèmes de génération de diagrammes en LaTeX notament TikZ et Pstricks, finamlement nous avons vu quelques exemples de dessin de differentes formes en utilisation le système TikZ.

Dans le prochain chapitre, nous allons présenter la conception du système permettant de générer des diagrammes LaTeX.

Chapitre 2 : Analyse et Conception

Introduction

Dans ce chapitre, nous allons présenter le processus de développement de notre application. Nous avons choisi le langage UML (Unified Modeling Language) en cause de sa qualité de modélisation orientée objet, de son formalisme relativement simple et clair qui permet de voir et de manipuler l'ensemble des éléments formant un système et sa rigueur qui est essentiel au bon déroulement du projet.

D'abord, nous présentons quelques définitions et concepts du langage UML. Par la suite, nous exposons la modélisation de notre système en utilisant les trois diagrammes les plus connus comme suivants :

1. Diagramme de Cas d'Utilisation.
2. Diagramme de Séquence.
3. Diagramme de Classe.

2.1 UML (Unified Modeling Langauage) :

UML (Unified Modeling Language), est un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. Les concepts des approches par objets : classe, instance, classification, etc…

UML unifie des méthodes objet, et plus particulièrement les méthodes Booch'93 de G. Booch, OMT-2 (Object Modeling Technique) de J. Rumbaugh et OOSE (Object-Oriented Software Engineering) d'I. Jacobson.

UML est une notation graphique conçue pour représenter, spécifier, construire et documenter les systèmes logiciels pour avoir une architecture logiciels. Il y a deux principaux objectifs :

- La modalisation des systèmes à l'aide des techniques orienter objet.
- La création d'un langage abstrait qui doit être : compréhensible par l'homme et Interprétable par la machine UML s'adresse à toutes les personnes chargées de la production, déploiement et du suivi de logiciels (analystes, développeurs, chefs projets, architectures) et il peut servir à la communication avec les clients et les utilisateurs du logiciels.

UML se compose d'une part des éléments de modélisation qui représentent toutes les propriétés du langage et d'autre part des diagrammes (de cas d'utilisation, de classes, d'objets, d'états-transitions, d'activités, de séquence, de collaboration, de composants et de déploiement) qui en constituent l'expression visuelle et graphique. Ces diagrammes peuvent être classés hiérarchiquement comme indiqué dans la figure suivante : [7]

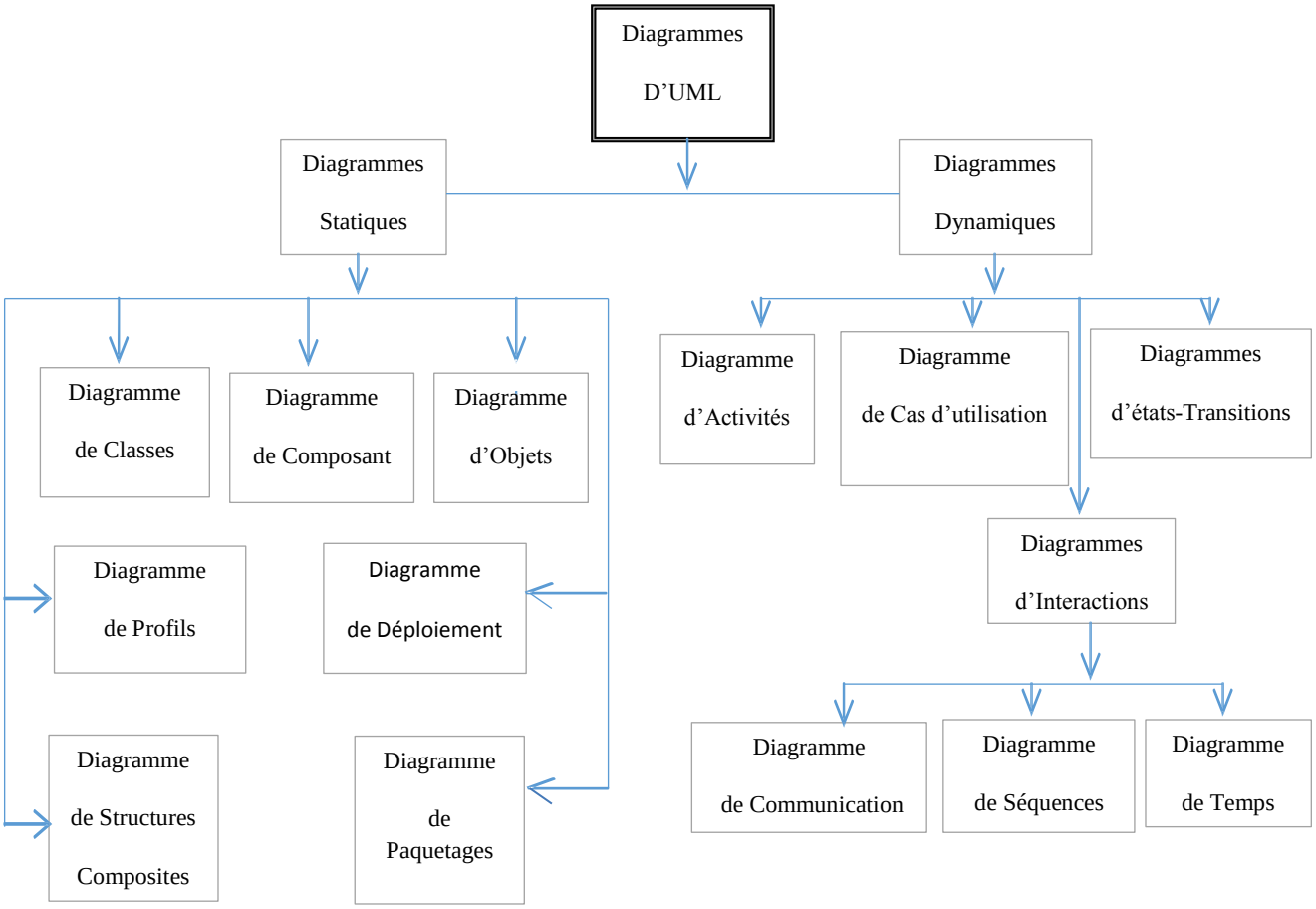


Figure 2.1: Classification des Diagrammes d'UML.

Pour la modélisation des besoins, nous utilisons les diagrammes UML suivantes :

2.2 Diagramme de Cas d'utilisation :

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobés par la limite du système, des associations de communication entre les acteurs et les cas d'utilisation, et des généralisations entre cas d'utilisation.

Il est destiné à représenter les besoins des utilisateurs par rapport au système.

2.2.1 Identification des Acteurs :

Les acteurs d'un système sont les entités externes à ce système qui interagissent avec lui. Dans notre application, le seul acteur qui interagit avec le système est l'utilisateur du système.

Il peut faire les taches suivantes :

1. Dessiner Shape
2. Déplacer Shape.
3. Ajouter libélé
4. Supprimer Shape.
5. Liée les différents Shapes par une ligne ou une flèche.
6. Ouvrir un Diagramme.
7. Enregistrer les diagrammes.
8. Imprimer un Diagramme.
9. Générer le code LaTeX d'un diagramme.
10. Copier le code LaTeX produit

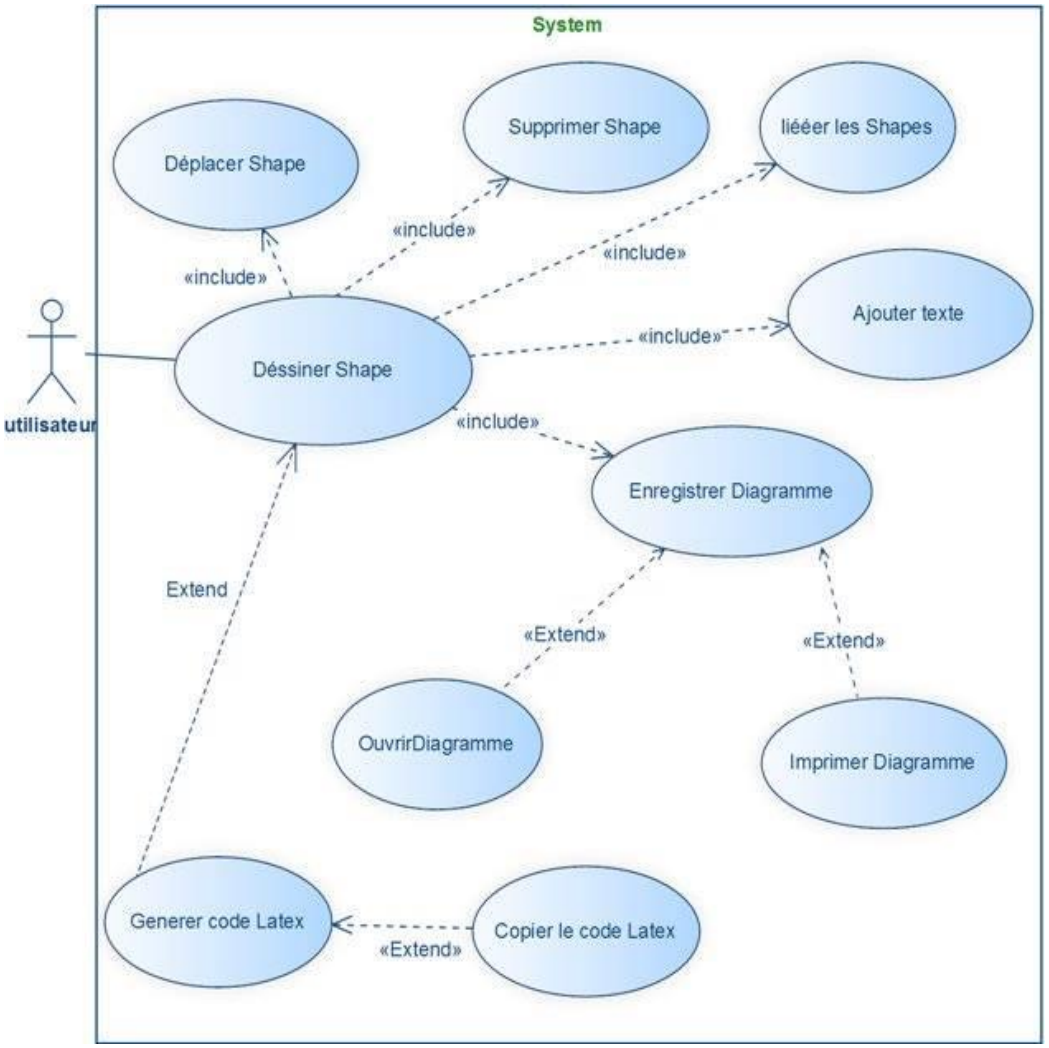


Figure 2.2: Diagramme de cas d'utilisation.

2.3 Diagramme de séquence :

Le diagramme de séquence montre les interactions entre les objets, agencées en séquence dans le temps ; il montre en particulier les objets participant à l'interaction par leurs lignes de vie et les messages qu'ils s'échangent ordonnancés dans le temps … mais il ne montre pas les relations entre les objets.[9]

Dans ce qui suit, nous présentons les interactions de l'utilisateur avec l'application à travers un diagramme de séquence.

A. Dessin Shape :

Lorsque l'acteur (utilisateur) décide à dessiner un Shape (une forme), il doit sélectionner le type de forme à dessiné et préciser le point de départ.

Aprés il peut ajouter un texte ou libélé.

Ce scénario est présenté par le diagramme suivant :

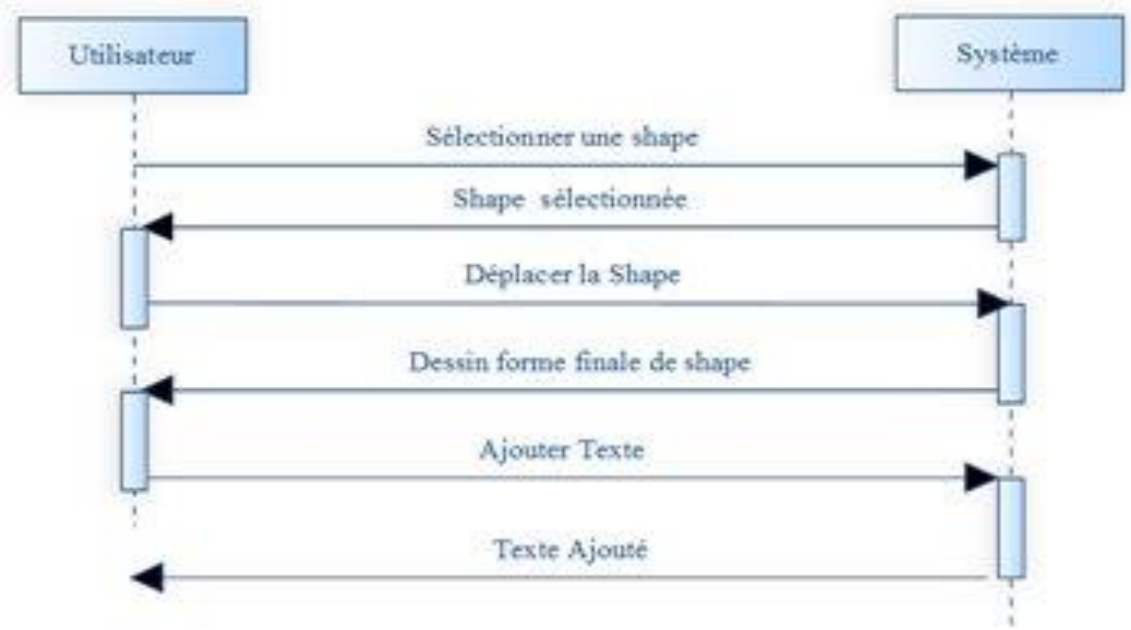


Figure 2.3: Diagramme de Séquence « Dessin Shape ».

B. Imprimer Diagramme :

L'utilisateur doit ouvrir un diagramme, puis il lance l'impression de ce diagramme.

Ce scénario est présenté par le diagramme suivant :

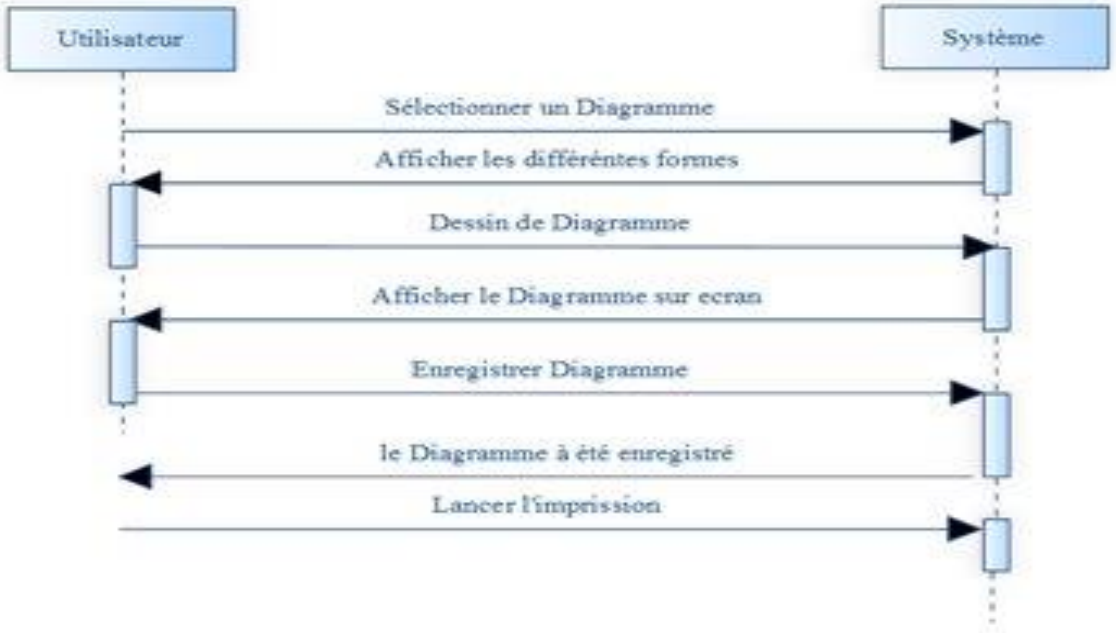


Figure 2.4: Diagramme de Séquence « Imprimer Diagramme ».

C. Générer le Code LaTeX:

L'utilisateur peut générer le code LaTex et le copier.

Ce scénario est présenté par le diagramme suivant :

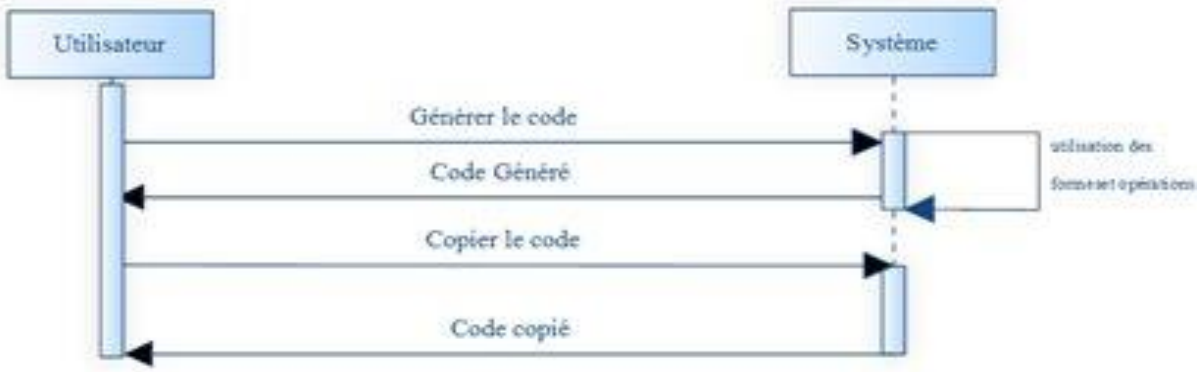


Figure2.5:Diagramme de Séquence « Générer le Code LaTeX ».

2.4 Diagramme de Classe :

Un diagramme de classes est une collection d'éléments modélisant la structure du système en faisant abstraction des aspects dynamiques et temporels. C'est à la fois l'axe essentiel de la modélisation objet et la notation la plus riche de tous les diagrammes UML. [7]

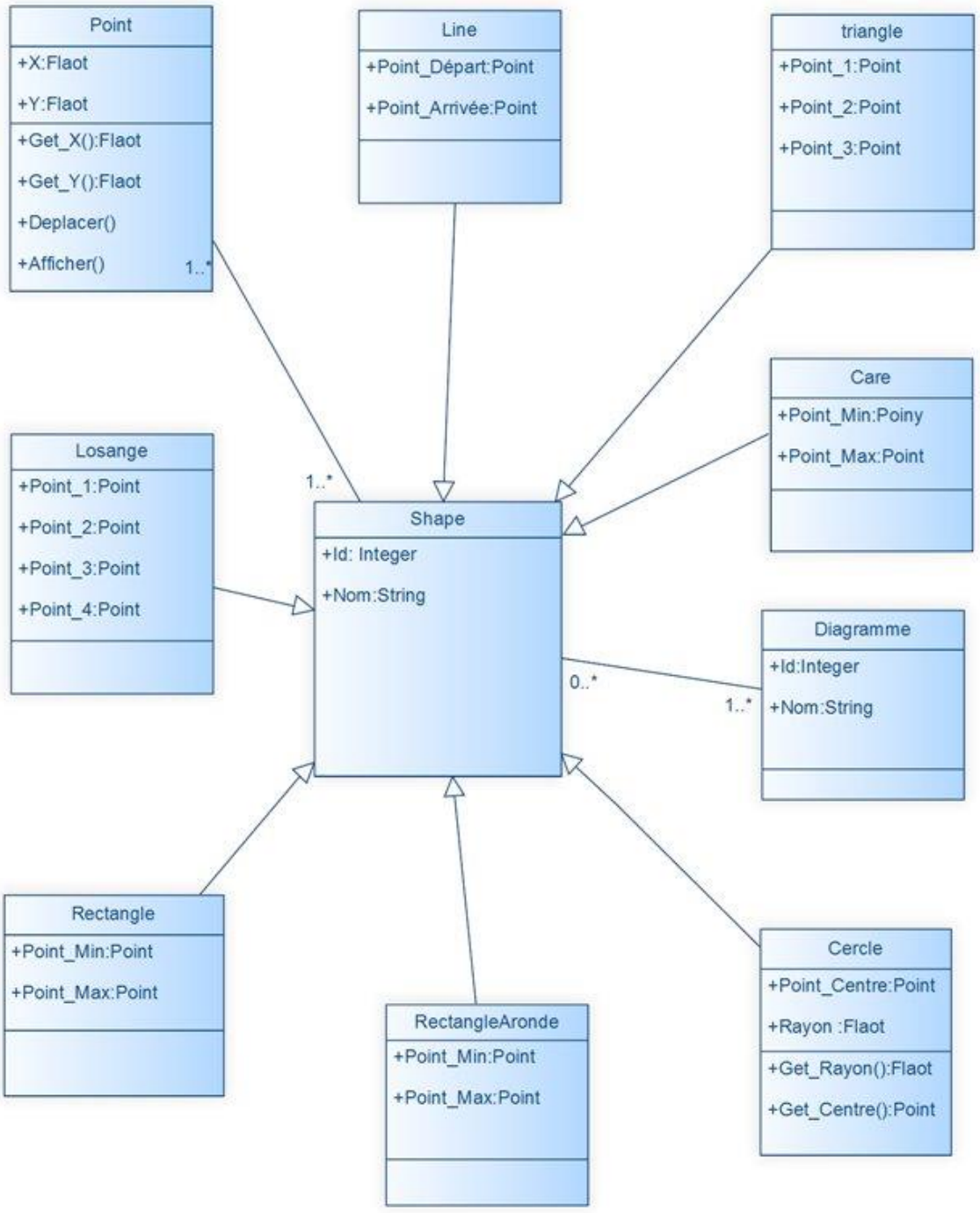


Figure 2.6: Diagramme de Classe du système.

Conclusion

Dans ce chapitre, nous avons présenté la modélisation de la structure statique et dynamique de notre système en utilisant un sous ensemble des diagrammes d'**UML**. Cette modélisation est une étape nécessaire et importante pour pouvoir créer et réaliser l'application. Dans le prochain chapitre, on détaillera l'application et ses différentes composantes.

Chapitre 3 : Réalisation

Introduction

Dans ce chapitre, on va détailler la réalisation et l'implémentation de notre application, on va voir également les outils utilisé pour développer ce travail qui génère un code LaTeX pour dessiner des formes ou diagrammes difficiles à faire avec LaTeX surtout pour un débutant.

A la fin on peut obtenir donc une application propre, facile à utiliser et à maintenir.

3.1 L'environnement de développement

Dans ce travail, nous avons choisis à programmer l'application avec langage C#. Dans les sections suivantes nous allons voir en bref quelques détails sur ce langage.

3.1.1 Langage de programmation (C #)

Notre application est développée par le langage de programmation C #, ce langage est un langage de programmation orienté objet (POO) a été développé par la société Microsoft , et notamment un de ses employés, Anders Hejlsberg, pour la plateforme .NET (point NET / dot NET). IL est très proche du langage Java.

Il est précompilé en MSIL (Microsoft Intermediate Language), puis exécuté sur une machine virtuelle, ou compilé en code natif à l'exécution. Il dispose d'un ramasse-miettes (garbage collector). Il utilise l'API .NET en remplacement des MFC (Microsoft Foundation Class). Il semble être le nouveau langage pour développer des applications Windows, avec Visual Basic et C++.

3.1.2 Quelques caractéristiques de C

Le langage C# possède des caractéritiques très intéressantes, parmi les on trouve :

- Compilation dans un langage intermédiaire indépendant de la machine et exécution dans un environnement dédié (une machine virtuelle)
- Gestion automatique de la mémoire grâce à un ramasse-miettes
- Introspection pour manipuler dynamiquement les objets
- Toutes les classes héritent d'une même classe (Object) et sont allouées sur le tas
- Pas de support de l'héritage multiple mais utilisation d'interfaces
- Tout doit être encapsulé dans une classe : il n'existe pas de fonctions ou constantes globales
- Gestion des erreurs grâce aux exceptions

3.1.3 Les points forts de C

- Le mode unsafe qui permet de manipuler la mémoire (hors du contrôle de l'environnement managé)
- Les propriétés, les indexeurs, les délégués et les événements
- La surcharge des opérateurs
- Les structures qui sont des types valeurs
- C# propose l'instruction goto

3.2 Le matériel utilisé pour réaliser l'application

Pour réaliser cette application on utilisant le matériel suivant :

— Laptop Dell.

— Processeur: Intel®Core (TM) i3-4005U CPU @ 1.70 GHZ.

— Ram 4.00 GO.

— Disque dur : 465.76 GO.

— Système d'exploitation : Windows 7 - 64bit.

3.3 Les logiciels Supplémentaires utilisés :

Nous avant utilisé des logiciels pour simplifer la tache de modélistion et de réalisation, à savoir :

— **Star UML 2.0.**

— **Visual studio 2017.**

3.4 Description de l'Application :

Dans notre application il y'a une seule interface principale, une fois l'utilisateur exécute l'outil il peut commencer à dessiner directement.

La figure 3.1 suivante présenté l'interface Principale de notre Application

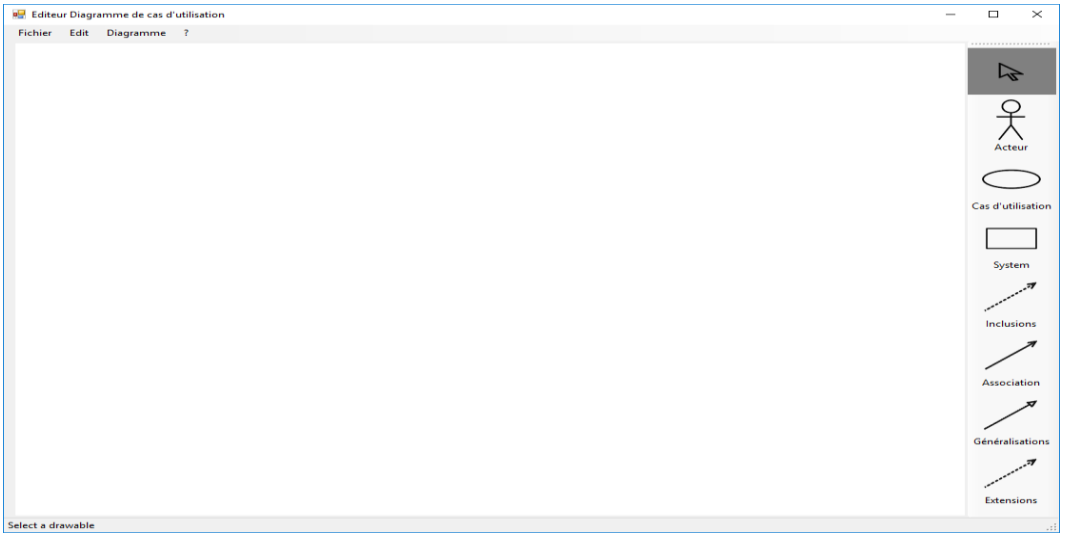


Figure 3.1: Interface Principal de l'Application.

3.4.1 Les Formes Disponible de l'Application :

L'utilisateur peut dessiner les différentes formes comme il est illustré dans la figure 3.2 suivante :

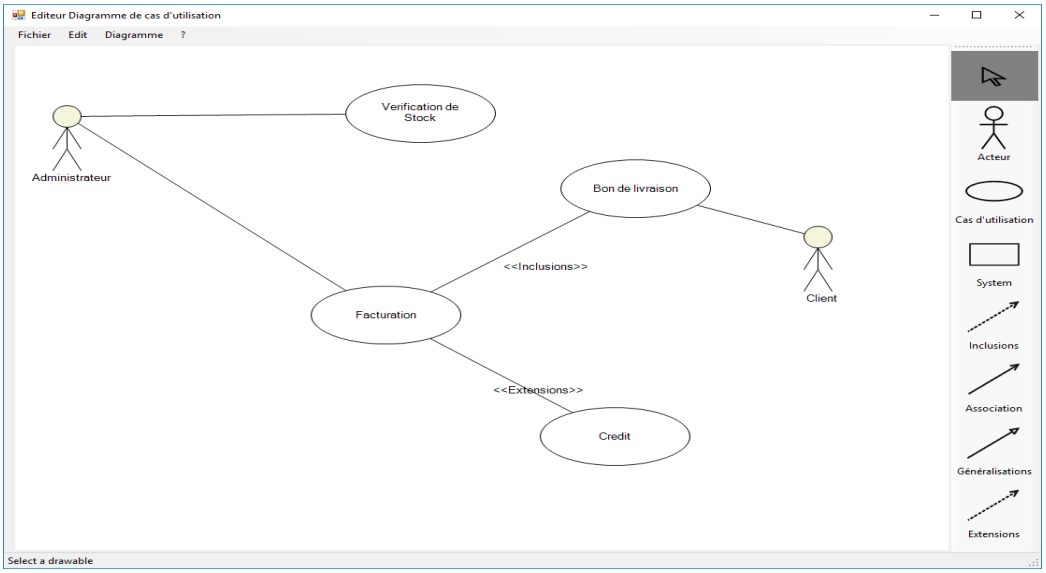


Figure 3.2: Les Formes Disponible de l'Application.

3.4.2 Les Opérations Sur la Shape :

L'utilisateur après avoir dessiné une shape il peut faire les opérations suivantes :

— Déplacer une shape.

— Supprimer une shape.

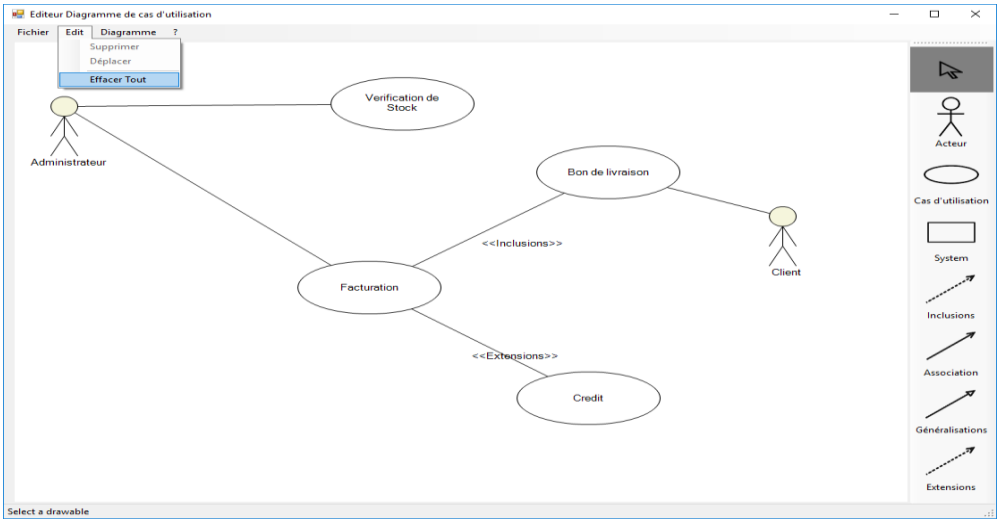


Figure 3.3: Les Opérations Sur la Shape.

3.4.3 Le Code LaTeX Après Génération :

L'utilisateur peut voir le code générer dans le deuxième onglet de l'interface principale comme ci de suite dans la figure 3.4.

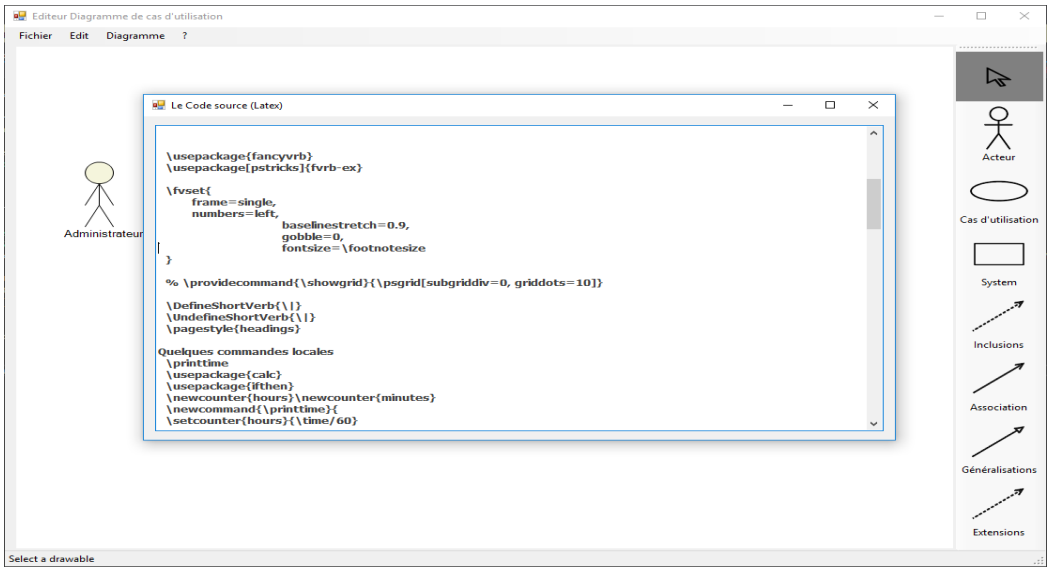


Figure 3.4: Le Code Après Génération.

3.4.4 La Barre de Menu de l'Application :

La figure 3.5 montre les différentes options de la barre de menu.

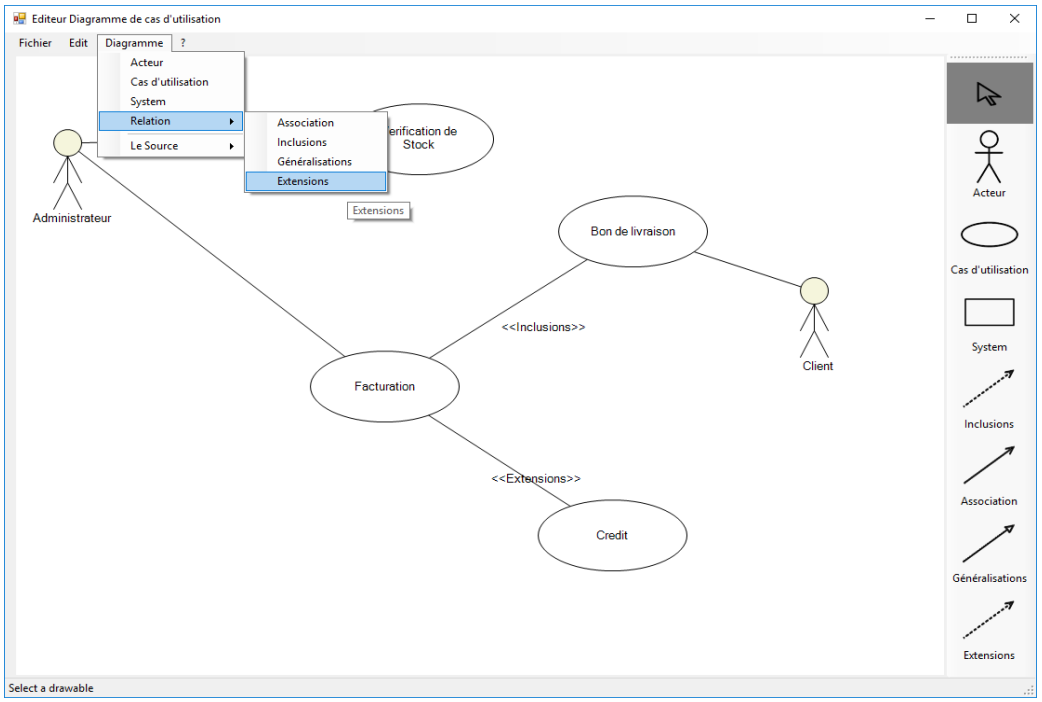


Figure 3.5: La Barre de Menu.

3.4.5 Comment Sauvegarder Le Diagramme :

Dans cette figure 3.6 L'utilisateur peut enregistrer le diagramme dessiné.

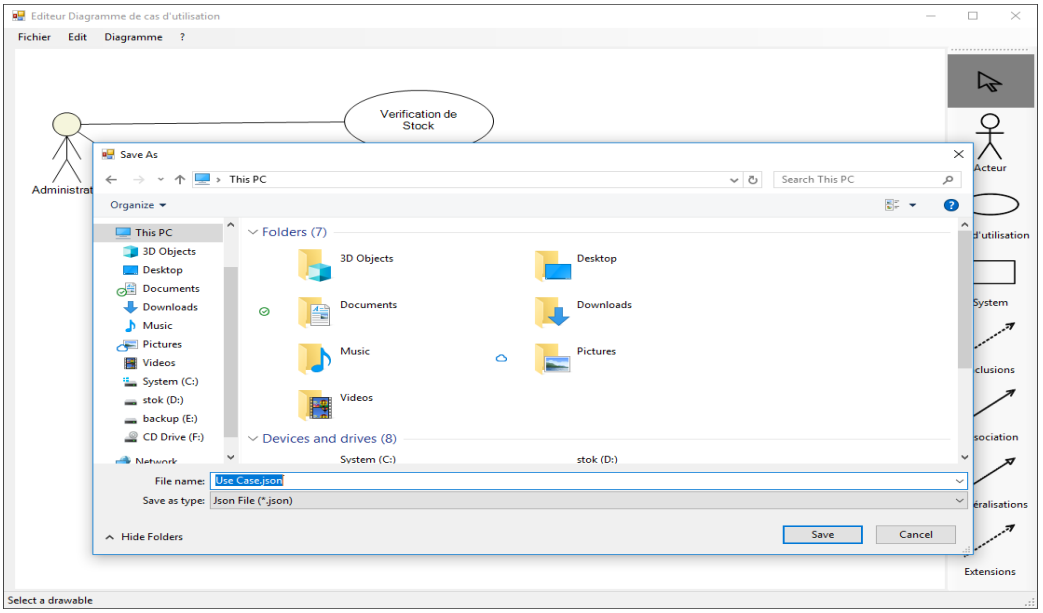


Figure 3.6: Sauvegarder Le Diagramm

3.4.6 Comment Ouvrir Le Diagramme :

L'acteur peut ouvrir un diagramme sauvegardé précédemment comme montre la figure 3.7 suivante.

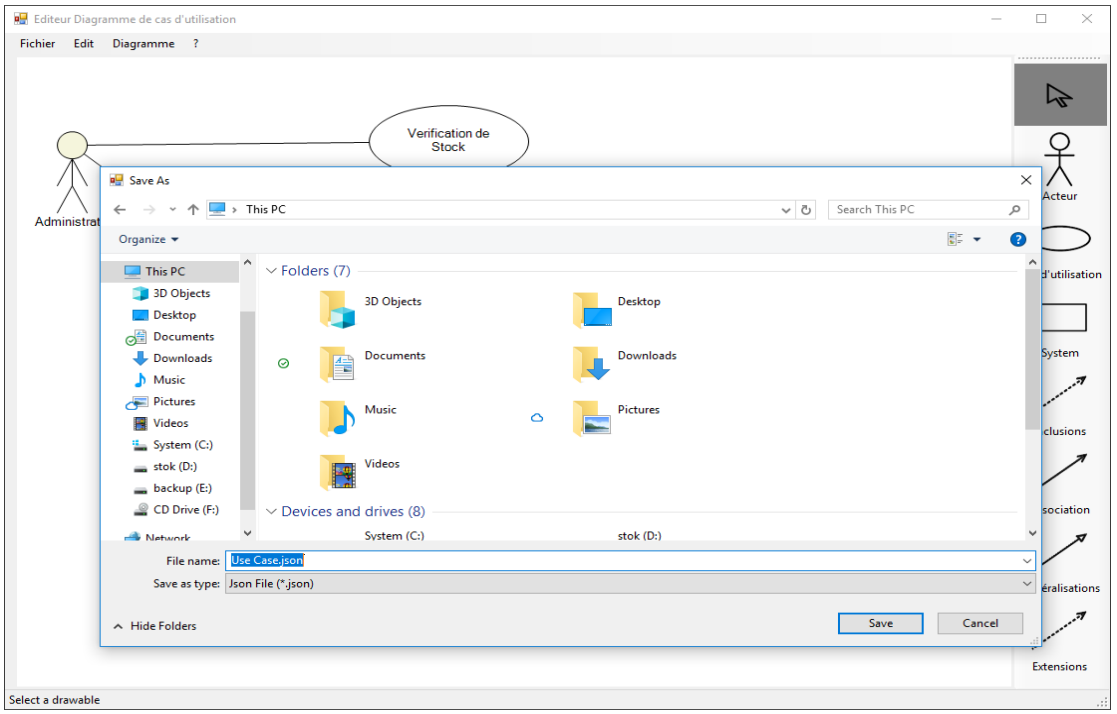


Figure 3.7: Ouvrir Le Diagramme.

Conclusion

Dans ce chapitre nous avons expliqué comment réaliser l'application et le langages utilisé et l'environnement de travail choisi pour développer notre projet, puis nous avons présenté les codes et les interface principales de notre logiciel.

Conclusion générale

L'objectif de notre travail est la réalisation d'un système permettant de simplifier la génération des diagrammes en LeTeX avec TikZ, tout en gardant la même qualité des diagrammes générés par commandes, et en minimisant l'erreur d'affichage graphique au maximum.

LaTeX est un métalangage de composition des documents scientifique d'une excellente qualité. Il utilise des commandes prédéfinies pour la génération des différents éléments (tableaux, figures, digrammes, . . . etc.). Par ailleurs, le codage des diagrammes en TikZ nécessite la connaissance et la maîtrise d'un nombre important de commandes et paramètres, ce qui est pas simple pour un débutant à cet environnement. De ce fait, il est très important d'avoir un outil de design des diagrammes LaTeX.

Cependant, pour passer de l'affichage du langage C # à l'affichage en TkiZ LaTeX il est nécessaire de résoudre les problèmes suivants :

1. C # utilise un système d'affichage en coordonnées d'écran autant que TikZ utilise un système d'affichage en coordonnées cartésien.
2. C # utilise un système de positionnement des formes à base de coin ou point supérieur gauche autant que TikZ utilise un système de positionnement des formes à base du centre.
3. Certains éléments sont dessinés en C# différemment qu'en TikZ, tel qu'un rectangle arrondi.

Pour pouvoir remédier à ces problèmes nous avons utilisés des fonctions de mapping afin de réaliser le passage avec une marge d'erreur graphique minimale.La correction de cette erreur nécessite d'utiliser le positionnement de LaTeX à base de direction : west, east, south et north.

Le travail que nous avons réalisé constitue une étape et un premier apport pour la génération des diagrammes de bonne qualité en LaTeX avec TikZ.

Bibliographie

[1] Baudoin, Marc,«Apprends LaTeX», Ecole nationale supérieure de techniques avancées,Paris, 1994

[2] Lucas GERIN, Romain PRIVAT, Yannick PRIVAT, «Petit guide pour les d´ebutants en LaTeX»,28 Avril 2008

[3] Thierry MASSON,«Cours 1 – Les fondamentaux : l'univers TeX»,Tech ,report,février2012

[4] Vincent, Lozano «Tout ce que vous avez toujours voulu savoir sur LA-TEX sans jamais oser le demander»,1st édition,In Libro Veritas,2008

[5] Présentation et exemples d'ulilisation LaTeX en sciences et en musique atelier B12 Christophe Pothier-Arruti 25 janvier 2017.

[6] Céline Chevalier, «Formation LaTeX – niveau débutant», Première partie, Mai-Juin 2009

[7] Laurent AUDIBERT, «UML 2», Institut Universitaire de Technologie de Villetaneuse– Département Informatique Avenue Jean-Baptiste Clément, 2007-2008

[8] TikZ pour l'impatient Gérard Tisseau Jacques Duma 11 février 2017

[9] Introduction à UML 2 Eric Cariou