

RAPPORT DU PROJET STL TIKZ

ANCELIN Maxime
DIAKHATE Aminata

Encadrant : PESCHANKSI Frédéric

3 juin 2012

Table des matières

1	Introduction	2
2	Présentation du projet	3
2.1	Contexte	3
2.1.1	Motivations	3
2.2	Objectifs	4
3	Gestion du projet	5
3.1	Choix technologiques	5
3.2	Difficultés	6
3.2.1	Installation de perlqt4	6
3.2.2	Identification des objets Tikz	6
3.3	Architecture	7
3.4	Gestion du temps	8
3.5	Résultats	10
4	Conclusion	20
5	Perspectives	21
6	Bibliographie	22
7	Annexe	23

1 Introduction

Dans le cadre de notre enseignement universitaire à l'UPMC(Université Pierre Marie Curie) en master Science et Technologie du Logiciel, nous devons réaliser un projet proposé par nos professeurs. C'est ainsi que nous avons choisi le sujet proposé par Monsieur Frederic Peschanski qui est de créer une interface graphique au dessus d'un sous-ensemble du langage Tikz, permettant la construction visuelle (plutôt que textuelle) des figures.

Nous allons tout au long de notre rapport vous expliquer comment nous avons gérer ce projet. Nous commencerons donc par vous faire une présentation générale de notre projet. Ensuite, nous vous parlerons de la gestion de notre projet ; et pour finir nous vous présenterons nos perspectives pour ce projet.

2 Présentation du projet

2.1 Contexte

LaTeX(Lamport TEX) est un langage et un système de composition de documents. Du fait de sa relative simplicité, il est souvent utilisé dans les domaines techniques et scientifiques pour la production d'un contenu complexe (équations, graphes, ...) ayant une mise en forme standard. Afin d'inclure ces contenus complexes dans les documents en restant dans l'environnement LaTeX, LaTeX dispose d'un package TikZ permettant d'inclure des figures au format PDF. En effet, TikZ permet d'obtenir des figures géométriques complexes précises et d'une grande qualité. Cependant son apprentissage n'est pas évident.

2.1.1 Motivations

Notre encadrant, qui est lui-même un utilisateur de TikZ, a souhaité faciliter l'utilisation de TikZ afin de permettre à un plus grand nombre d'utilisateurs de profiter de ses avantages sans forcément passer par son apprentissage. Pour ce faire, il nous propose donc de mettre en place une interface graphique permettant la construction visuelle (et non textuelle) des figures.

Pour ce projet que nous devons réaliser dans le cadre de notre master, nous avons souhaité, avant même de connaître les sujets proposés, de choisir un projet qui nous permettrait de nous rapprocher le plus de ceux qu'on pourrait rencontrer dans le monde professionnel. Par la suite après avoir étudié tous les sujets proposés, ce projet a retenu notre attention parce qu'il nous permettait d'approfondir nos connaissances tout

en réalisant un projet qui diffère de nos projets habituels.

2.2 Objectifs

Notre objectif est de pouvoir présenter au terme de ce projet une interface graphique qui permettrait à l'utilisateur de pouvoir créer des graphes non plus textuellement mais visuellement. L'interface graphique devra aussi permettre à l'utilisateur de rédiger du code TikZ et visualiser le graphe correspondant ou encore modifier le graphe en bougeant les noeuds sélectionnés ou en modifiant leurs propriétés grâce à un menu. Toute modification apportée au graphe devra mettre à jour le code source correspondant et vice-versa.

3 Gestion du projet

3.1 Choix technologiques

Pour la réalisation de notre projet, notre encadrant nous a recommandé l'utilisation d'un langage de script tel que Python ou Ruby, permettant l'interfaçage avec des outils externes (notamment pdflatex avec le paquet "preview") et les manipulations textuelles simples (expressions rationnelles etc.), et proposant de plus des bibliothèques portables pour les interfaces utilisateurs (Gtk, Qt, WxWindows, etc.). Même si notre encadrant nous a vivement conseillé l'utilisation de Python nous avons quand même choisi de faire notre projet avec Perl.

Perl nous a semblé être un bon choix de langage car il est adapté au traitement et manipulation de fichiers texte du fait de l'intégration des expressions régulières dans la syntaxe même du langage or nous voulions manipuler des fichiers LaTeX et Tikz ; de plus perl est portable sur la plupart des systèmes d'exploitation, il devient même populaire sous Windows grâce à la distribution gratuite ActivePerl. Pour l'interface utilisateur nous avons choisi la bibliothèque Qt comme nous le conseillait notre encadrant. Au final nous avons donc utiliser PerlQt qui allie la souplesse, la puissance et le design élégant de Qt au langage Perl. Pour l'installater, nous avons dû choisir entre deux versions de PerlQt4 :

- Une version avec une vue sur Qt-sys.org publié par l'utilisateur "vadiml" et qui n'a pas été mise à jour depuis Février 2008
- Une autre avec une vue sur code.google.com publié par "chrisburel@gmail" qui semblait plus complète et surtout activement maintenue puisque la dernière version datait de quelques jours seulement(février 2012).

Notre choix c'est donc naturellement porté sur la version proposée par "chrisburel@gmail".

Pour notre interface graphique, nous avons besoin d'un éditeur qui nous permettrait d'écrire du code Tikz. Pour cela nous avons choisi Scintilla qui est un composant d'édition de code open source. En effet Scintilla a beaucoup de fonctionnalités qui rendent l'édition de code plus facile telles que la coloration syntaxique, la gestion des numéros de ligne dans la marge, les points d'arrêts pour le débogueur, etc.

Au début de notre projet nous n'avions pas utilisé de gestionnaire de version car nous travaillions le plus souvent ensemble. D'ailleurs à la première semaine de notre projet nous avons réalisé le mockup de notre interface avec Michael MENU, un étudiant de notre promotion qui avait le même sujet que nous mais développé en Python. Pendant les vacances scolaires du mois d'avril où la distance ne nous permettait pas de continuer à travailler ensemble, nous avons installé git qui est un logiciel de gestion de versions décentralisé libre.

3.2 Difficultés

3.2.1 Installation de perlqt4

L'installation de perlqt4 nous a pris un peu plus d'une semaine, car il nous manquait des dépendances.

Au fur et à mesure que nous découvrons ce qu'il fallait ajouter, nous avons enrichi un script d'installation afin de rendre la réinstallation plus facile pour les utilisateurs finaux de notre application. Ce script a été conçu pour fonctionner sous *Ubuntu 12.04* et devrait marcher sans encombre sous *Debian*.

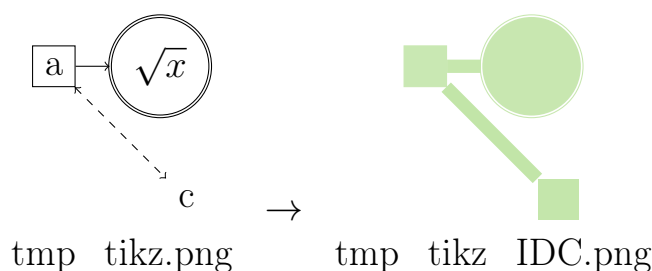
Sous ces *Linux*, il suffit de lancer **script_intall_all.sh** afin d'installer *perlqt4*, *Scintilla* et tous les packages et modules *Perl* requis au fonctionnement de l'application.

3.2.2 Identification des objets Tikz

Afin d'identifier indépendamment chaque objet Tikz, nous attribuons à chaque nœud et à chaque arête du graphe une couleur unique, que nous nommerons ColorID.

N'Y A T-IL PAS UNE REPETITION DANS CES 2 PARAGRAPHES ? Cette couleur sera ensuite ajoutée aux propriétés de chaque objet, puis on générera une image où chaque objet sera intégralement colorié de cette couleur. Nous nommerons cette image *tmp_tikz_IDC.png*

Rappelons que dans le cadre de notre projet nous ne nous intéressons qu'aux nœuds et arêtes (notés **node** et **draw**). L'attribution d'un code couleur à chaque objet identifié nous permet de générer une image temporaire où chaque élément est intégralement coloré en sa couleur.



Chacuns des objets de *tmp_tikz_IDC.png* a une couleur différente, la variation est juste très légère.

Ajout du colorId aux objets Tikz

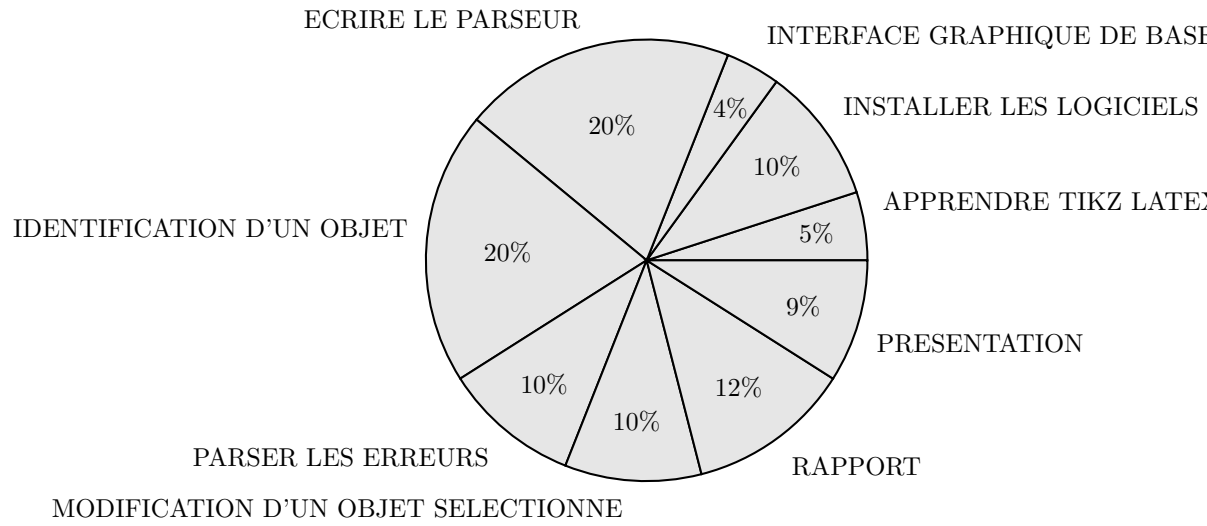
Tout d'abord, donnons un bref aperçu de Tikz. Chaque objet Tikz possède un champ de propriétés. Dans le cas de la couleur et de la couleur de remplissage (*fill*), la dernière valeur attribuée est la valeur finale. On peut donc intégralement colorier un objet d'une seule couleur, par exemple en bleu, en ajoutant à la fin des propriétés *blue,fill=blue*. On peut avoir une plus grande variété de couleurs que celles prédéfinies en utilisant le package *xcolor*, qui permet de donner un pourcentage à une couleur et même de les mélanger. Par exemple : *blue!50!red!13* Nous avons décidé de générer

3.3 Architecture

DIAGRAMME DES CLASSES EN MOINS DETAILLE

3.4 Gestion du temps

Le graphe ci dessous illustre le temps réparti à chacune de nos principales tâches. Sachant que 100% représente le temps total passé sur le projet.



3.5 Résultats

Nous allons à présent vous présenter en images les résultats de notre projet .

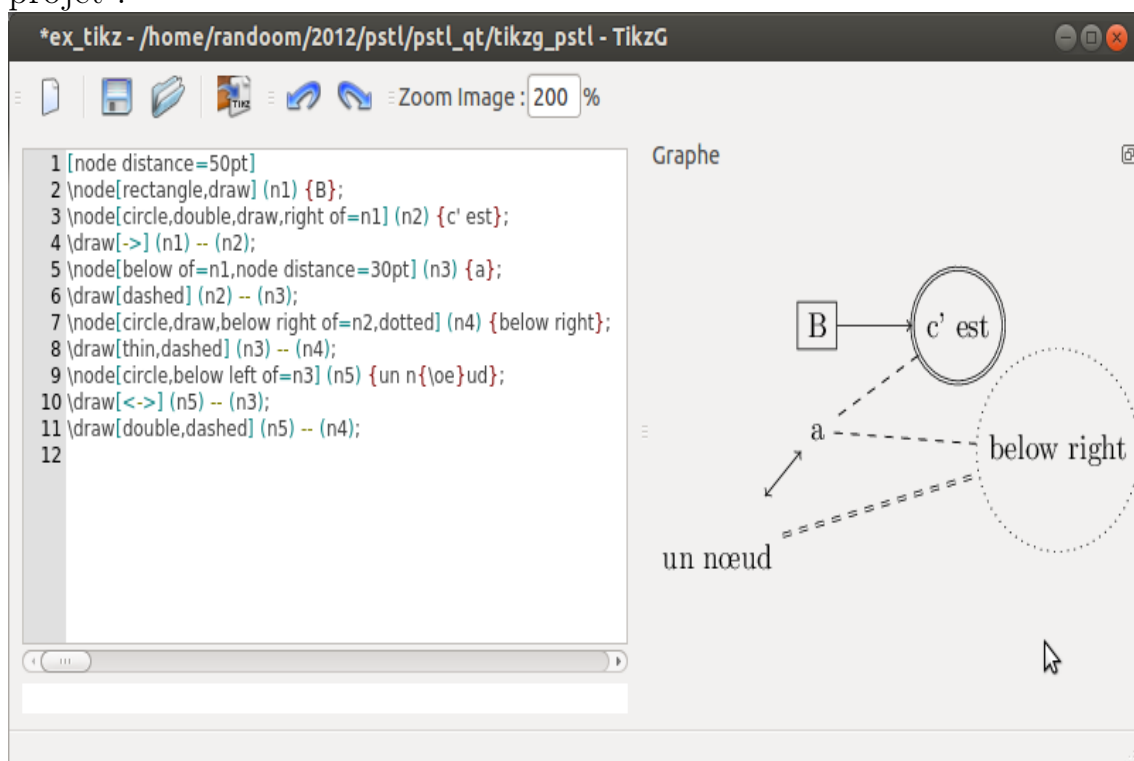


figure 1 : génération du graphe à partir du code tikz édité

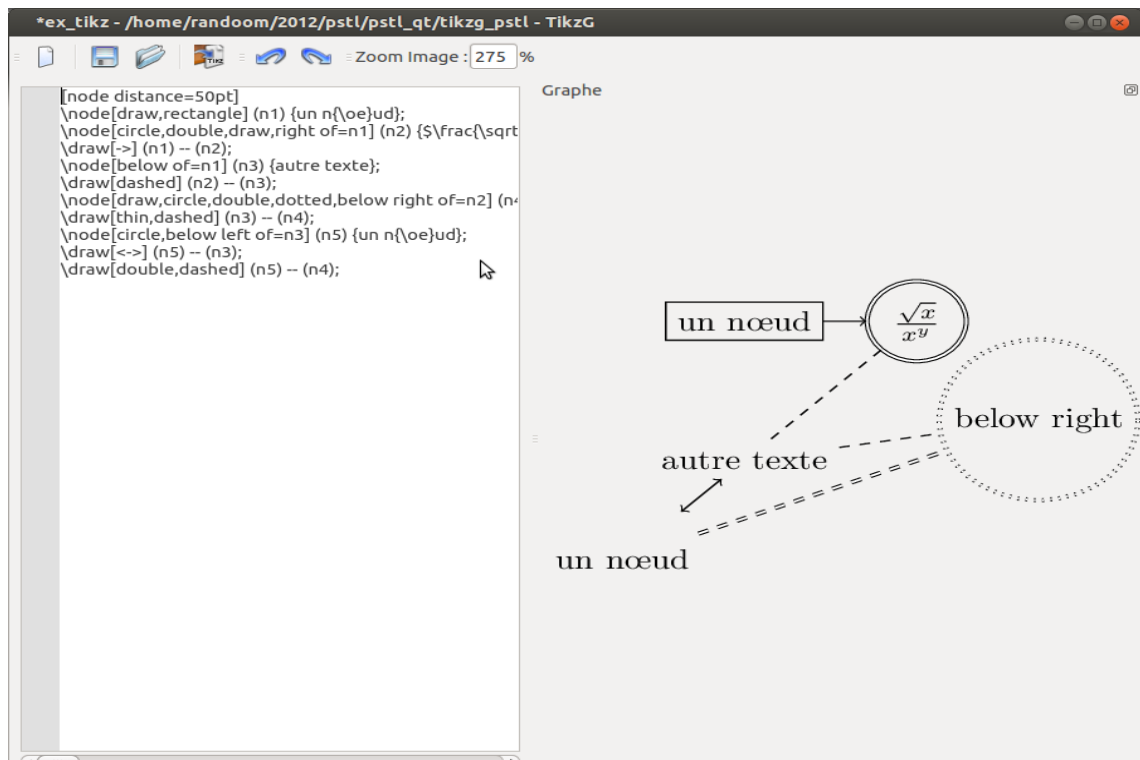


figure 2 : zoom et désactivation de la coloration syntaxique et des numéros de ligne dans la marge

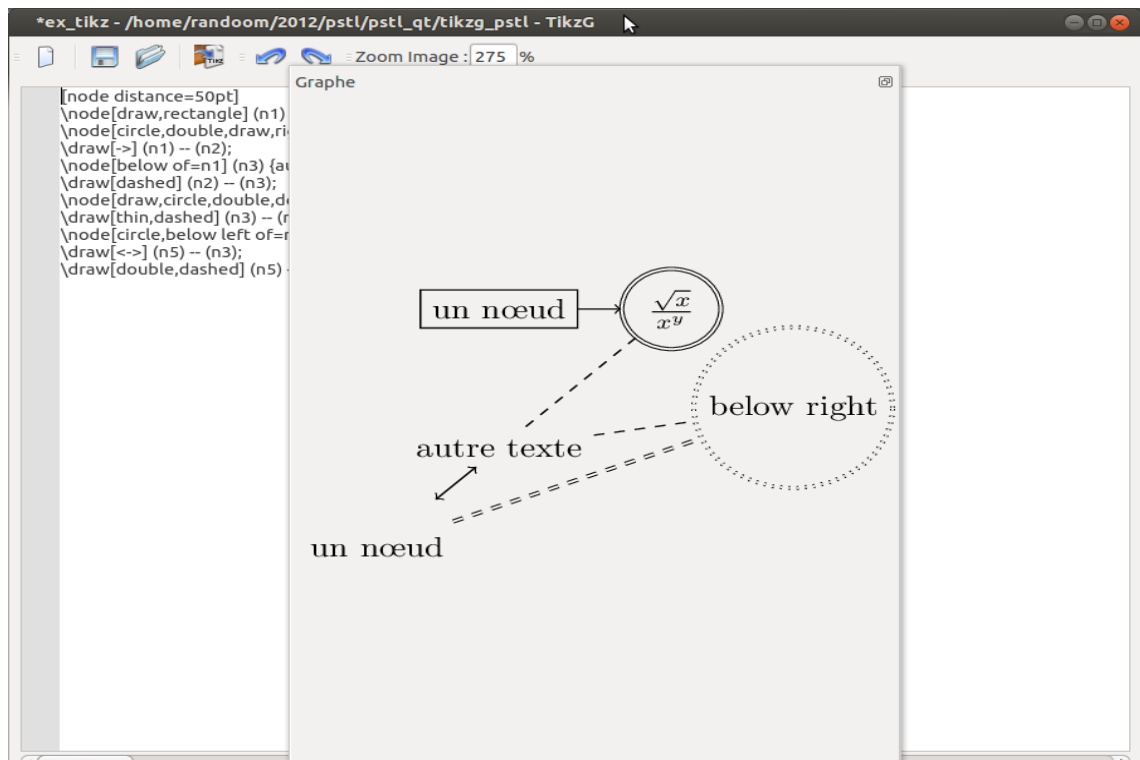


figure 3 : fenêtre du graphe flottante

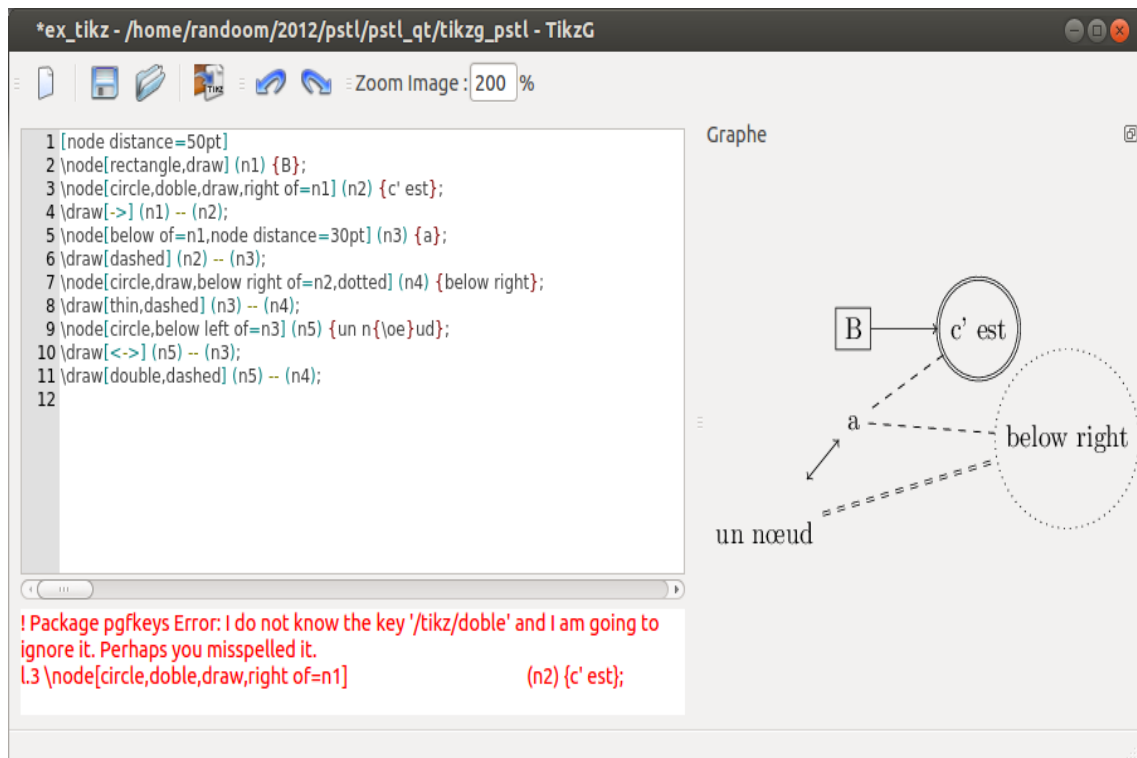
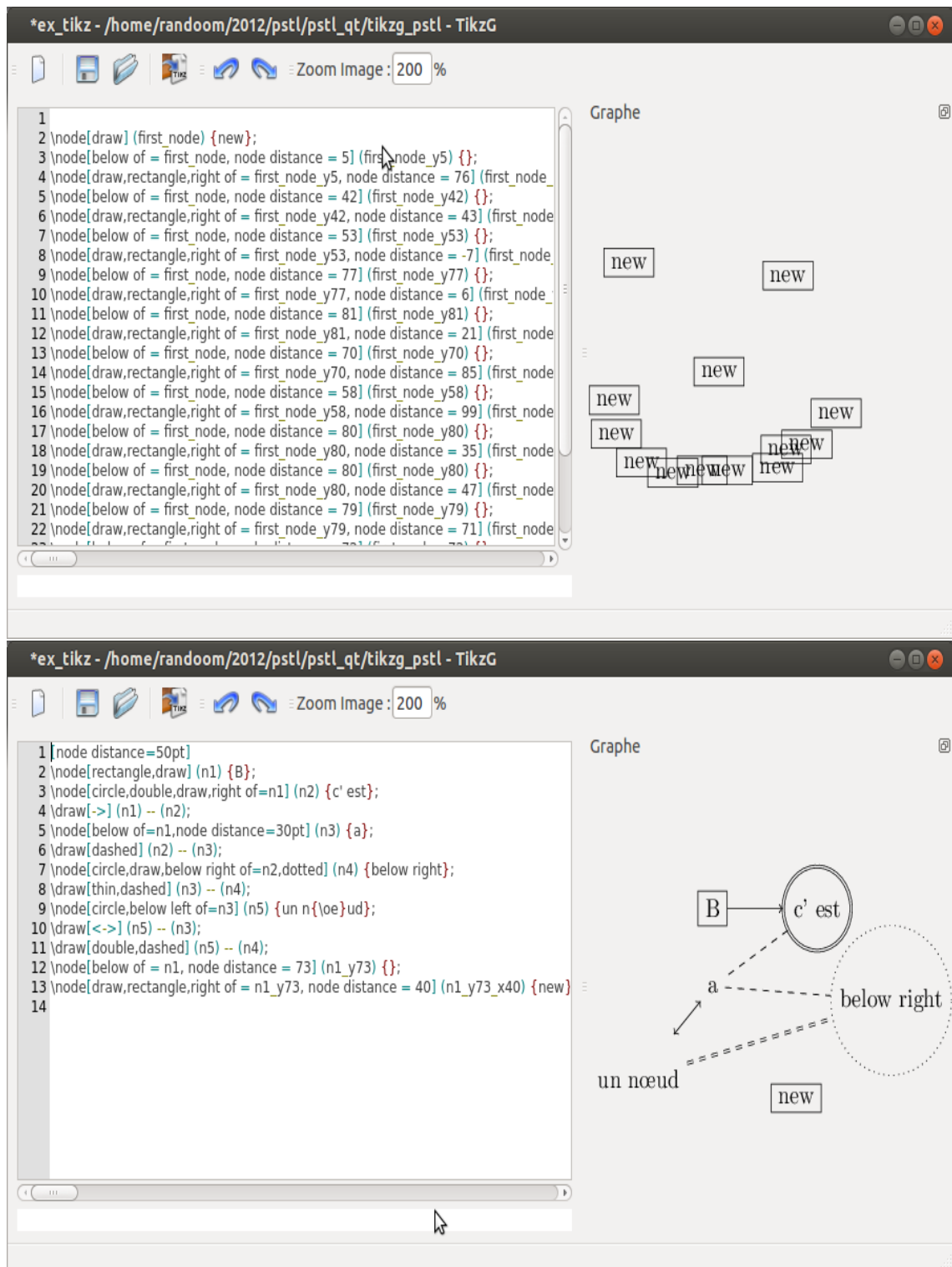


figure 4 : gestion d'erreurs

Comme nous pouvons le constater sur la figure 1, le graphe est généré à partir du code Tikz écrit dans l'éditeur de texte. Cette génération est relancée automatique au bout d'une seconde d'inactivité après modification du code source. L'utilisateur pourra choisir de cacher les numéros de ligne ou de désactiver la coloration syntaxique en décochant les cases correspondantes dans la barre de menu ; il pourra aussi définir la valeur du zoom dans la barre d'outil ou zoomer avec la molette de la souris. En effet, sur la figure 1 on peut noter l'affichage des numéros de ligne, la coloration syntaxique du code et la valeur du zoom qui est de 200% alors que sur la figure 2 l'affichage des numéros de ligne est désactivée de même que la coloration syntaxique et la valeur du zoom vaut 275% après un zoom avec la molette de la souris.

De plus, comme l'atteste les figures 3 et 4, les erreurs sont gérées et la fenêtre du graphe peut être détachée de l'interface et déplacée sans que cela ne modifie ses fonctionnalités.



figures 5 : création graphique d'un nœud

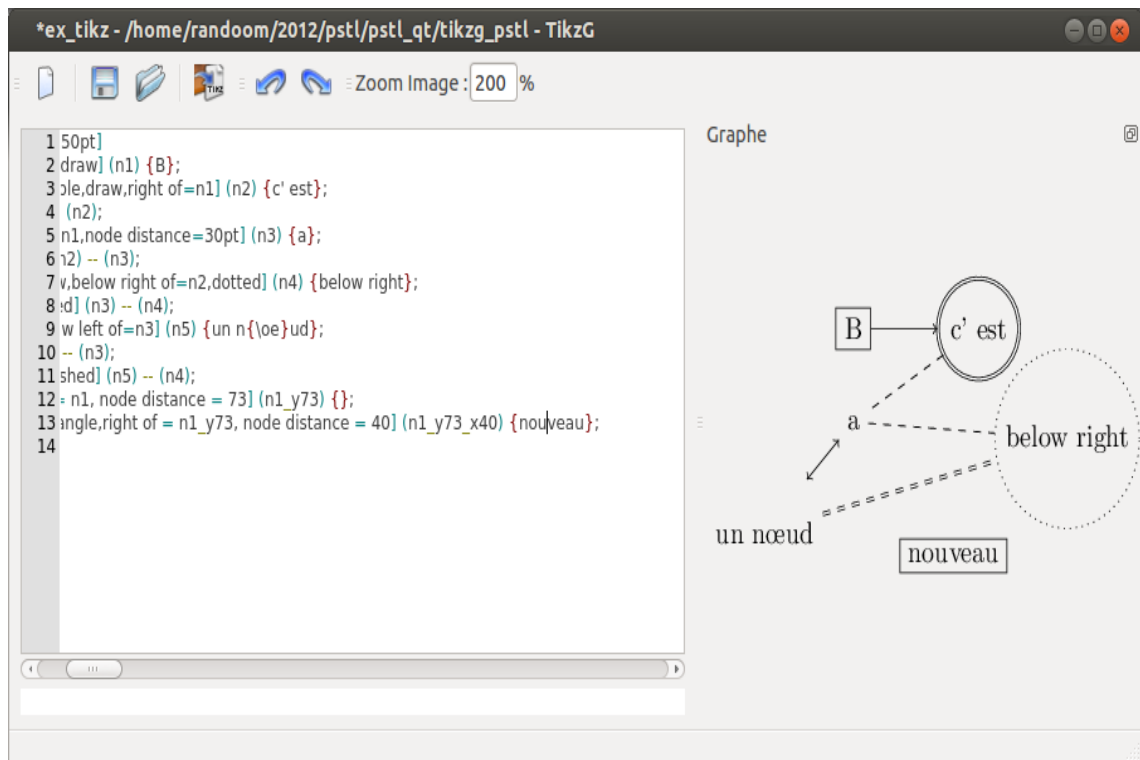
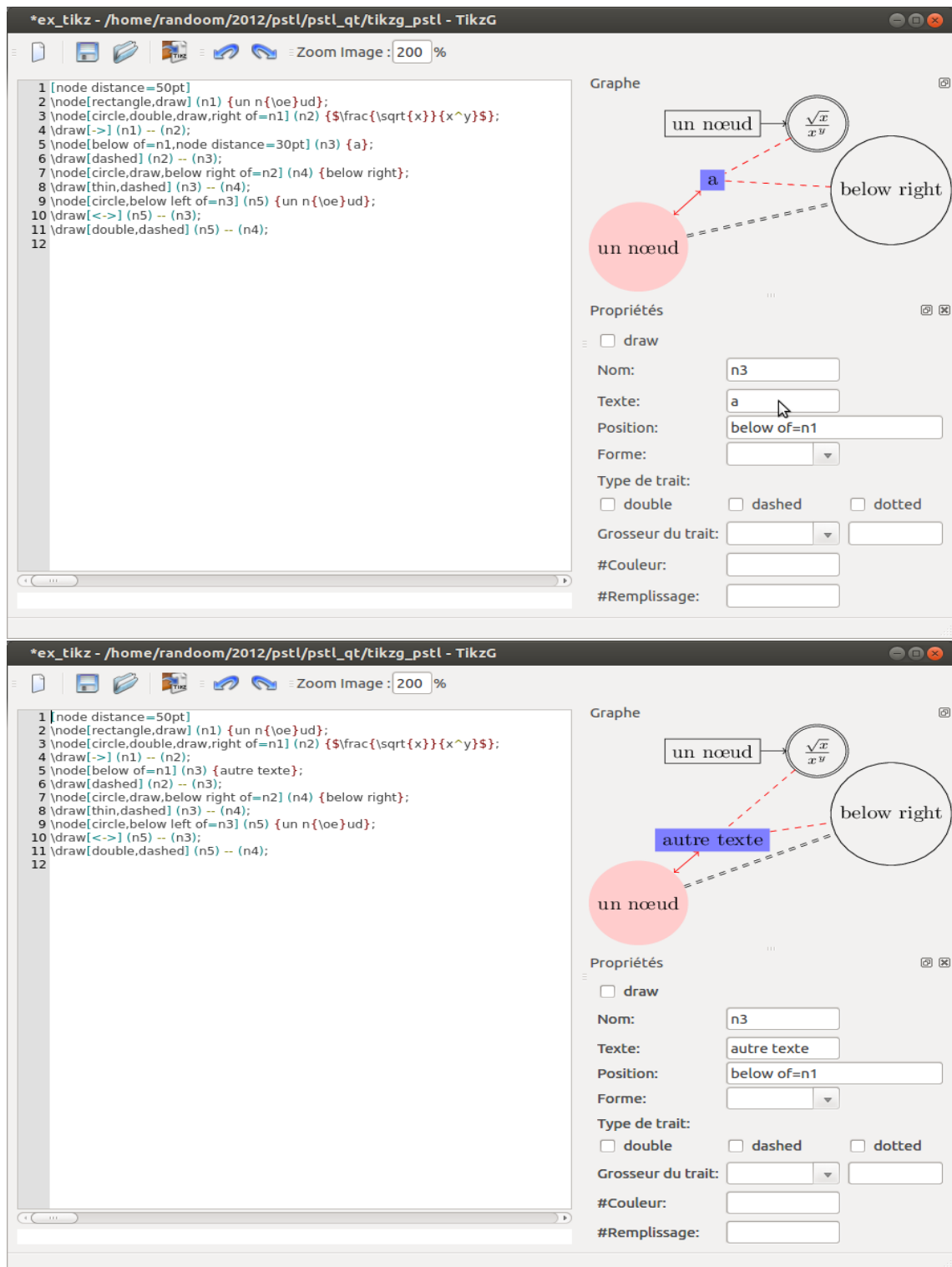


figure 6 : modification textuelle des propriétés d'un nœud

Un clic droit sur l'interface graphique permet de créer un nœud à cet endroit. L'utilisateur pourra par la suite modifier textuellement ou visuellement les propriétés de ce nœud. Le code correspondant au nœud ainsi créé est automatique ajouter au code du graphe.

Par exemple sur la figure 5, le nœud contenant le texte "new" a été créé par un clic droit de la souris, et les lignes 12 et 13 qui correspondent au code Tikz de la création de ce nœud ont été ajouté à l'éditeur de texte. Après une modification sur le code du texte de ce nœud(new qui devient nouveau), on observe sans surprise que le texte a aussi changé sur le graphe.



figures 7 : modification graphique des propriétés d'un nœud

Lorsqu'on clique sur un objet du graphe ses propriétés s'affichent. Prenons comme exemple les figures 7, on voit que lorsqu'on clique sur le noeud n3, il est coloré en bleu, ses noeuds relatifs en rose et les arêtes auxquelles il est relié en rouge. A partir de sa fenêtre de propriétés on peut apporter des modifications au noeud ainsi sélectionné. Et comme on peut le remarquer sur ces figures, le texte "a" contenu dans ce noeud devient "autre texte" après une modification à partir de la fenêtre des propriétés.

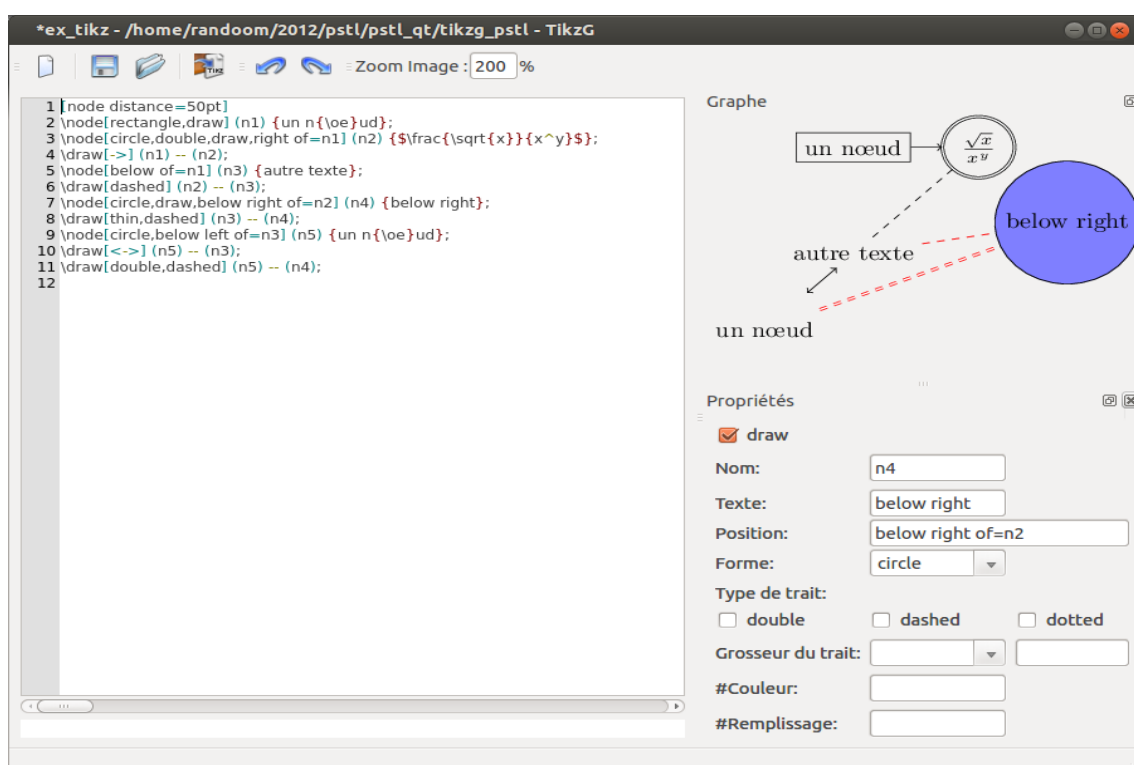


figure 10 : sélection d'un noeud du graphe

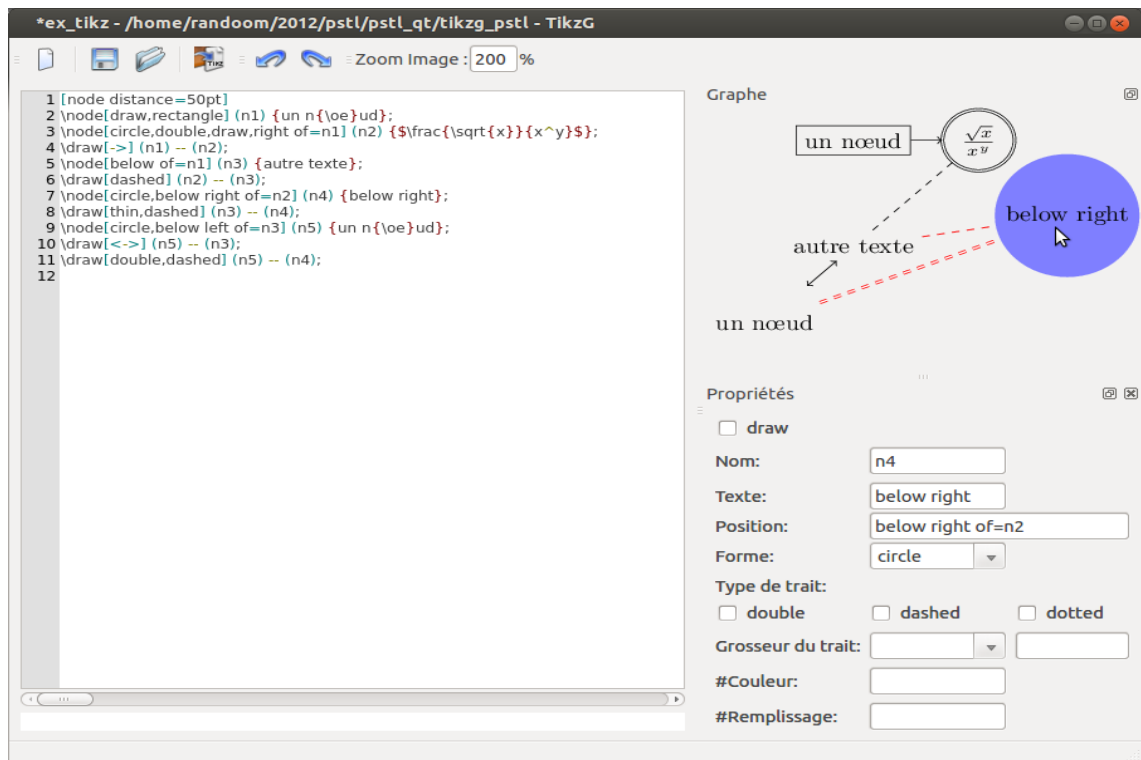


figure 11 : modification d'un nœud du graphe

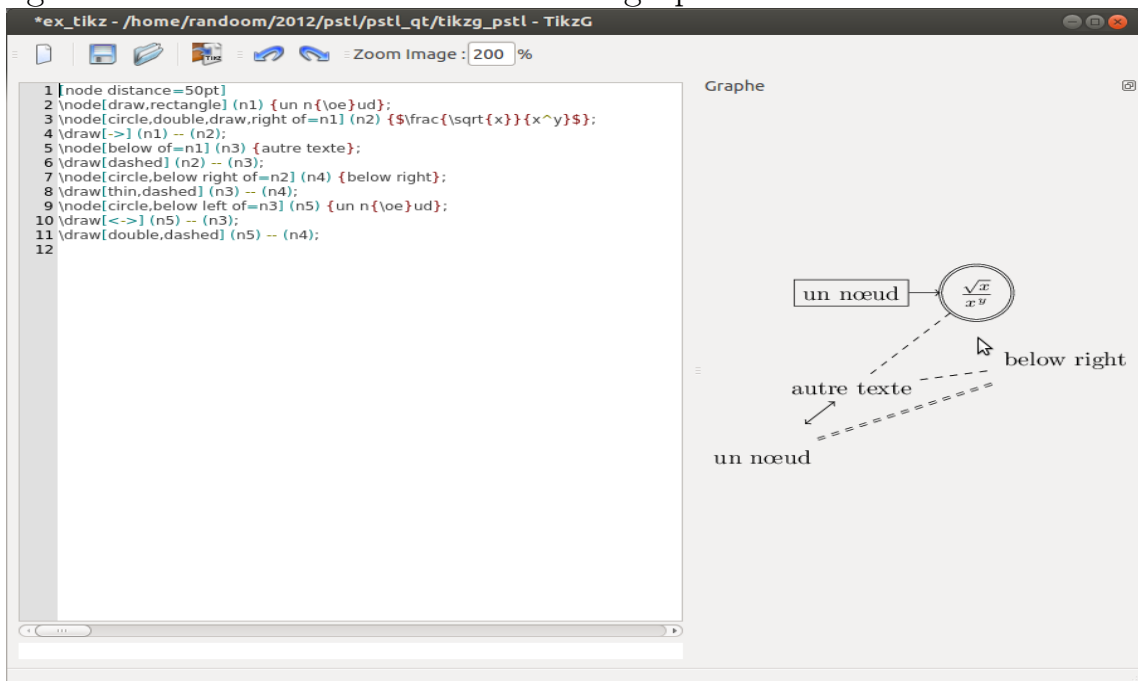
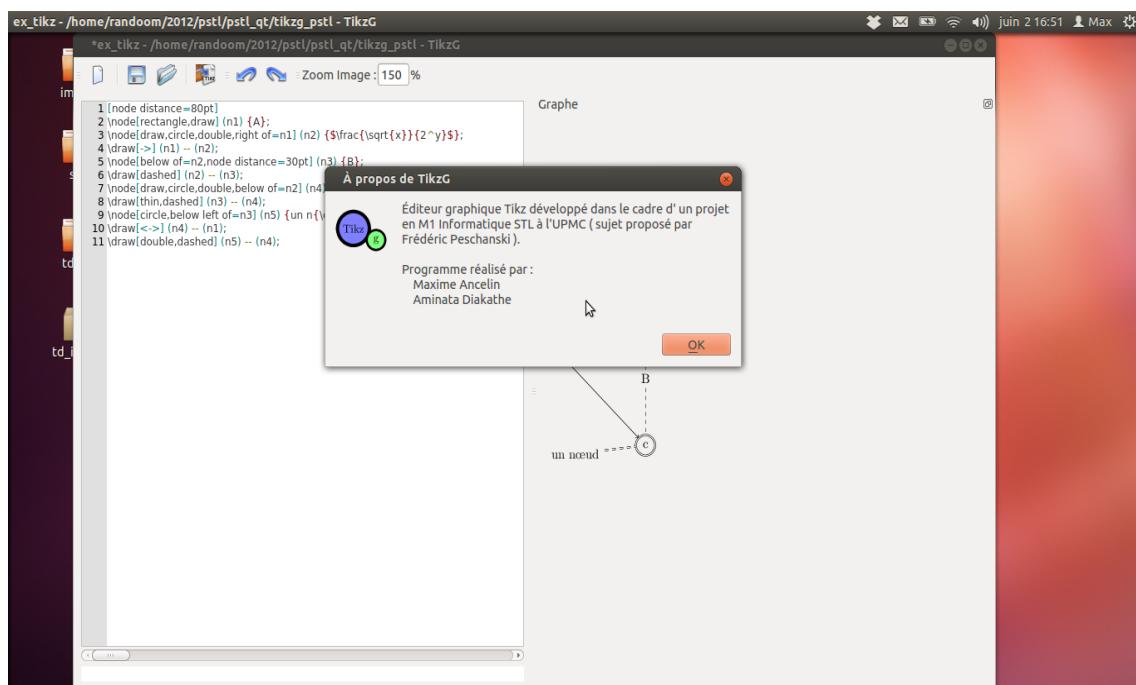


figure 12 : après modification d'un nœud du graphe

Les figures 10, 11 et 12 illustrent bien nos propos ci-dessus. On peut observer que sur la figure 10 dans la fenêtre Propriétés, la case draw qui correspond au dessin du nœud est cochée après l'avoir décoché à la figure 11 on voit effectivement que, sur la figure 12, le noeud n'est plus dessiné et la propriété "draw" a disparu à la ligne 7 du code Tikz.



figures 12 : à propos

4 Conclusion

Nous avons créé une interface graphique permettant d'éditer du code Tikz, de visualiser l'image généré et de modifier visuellement les propriétés des nœuds dans le cadre de notre projet Science et Technologie de Logiciel. En effet, avec notre interface graphique l'utilisateur pourra désormais créer graphiquement des nœuds, modifier les propriétés des objets du graphe sélectionnés tout en récupérant le code Tikz correspondant à la création de ce graphe. Il pourra aussi charger directement un fichier Tikz dans l'éditeur afin de visualiser et/ou modifier le graphe. Le code Tikz édité pourra ainsi être sauvegardé de même que le fichier pdf généré par le code Latex englobant le code Tikz.

Ce projet nous a permis d'appliquer les connaissances qui nous ont été inculquées au cours de nos années de Licence et de Maîtrise Informatique à l'Université Pierre et Marie Curie. Nous avons été confrontés à de nombreux problèmes et dans la plupart des cas nous avons pu trouver une solution alternative afin de les résoudre.

Enfin ce projet aura été l'occasion de découvrir et d'utiliser des outils dont nous n'avions pas la moindre idée de leurs existences.

5 Perspectives

Au vu des fonctionnalités de notre interface, nous pouvons conclure que les principaux objectifs du projet sont atteints. Toutefois, afin de faciliter la distribution de notre logiciel, nous aurions pu créer un paquet Debian et un exécutable Windows. De même pour une amélioration future de l'application, la gestion de la modification visuelle de la position d'un noeud pourra être réalisée. Cependant, le temps et la complexité de ces tâches ont été les principaux facteurs qui nous ont rebuté.

6 Bibliographie

Introduction à Perl

Auteur(s) : Randal L. Schwartz , Tom Phoenix

Editeur : O'Reilly

De l'art de programmer en Perl

Auteur(s) : Damian Conway

Editeur : O'Reilly

Perl Cookbook :

Auteur(s) : Christiansen

Tikz pour l'impatient (livre en français) :

<http://math.et.info.free.fr/TikZ/bdd/TikZ-Impatient.pdf>

Introduction à Latex :

<http://mirrors.ctan.org/info/lshort/french/lshort-fr.pdf>

7 Annexe

Manuel d'utilisation du logiciel TIKZG

ANCELIN Maxime
DIAKHATE Aminata

Encadrant : PESCHANKSI Frédéric

3 juin 2012

0.1 Avant propos

0.1.1 Qu'est ce que le logiciel TIKZG ?

TIKZG est une application permettant la création et la modification de graphes. Il est accessible de tout poste de travail sous Ubuntu.

0.1.2 Qu'offre le logiciel TIKZG ?

- Il permet d'éditer du code Tikz. L'utilisateur pourra ainsi visualiser le graphe généré.
- Il signale à l'utilisateur les éventuels erreurs du code en précisant la ligne concernée.
- Il permet à l'utilisateur de modifier graphiquement le résultat d'un code Tikz chargé dans l'application.
- Il permet à l'utilisateur d'enregistrer les modifications apportées au code source après modifications visuelles du graphe.

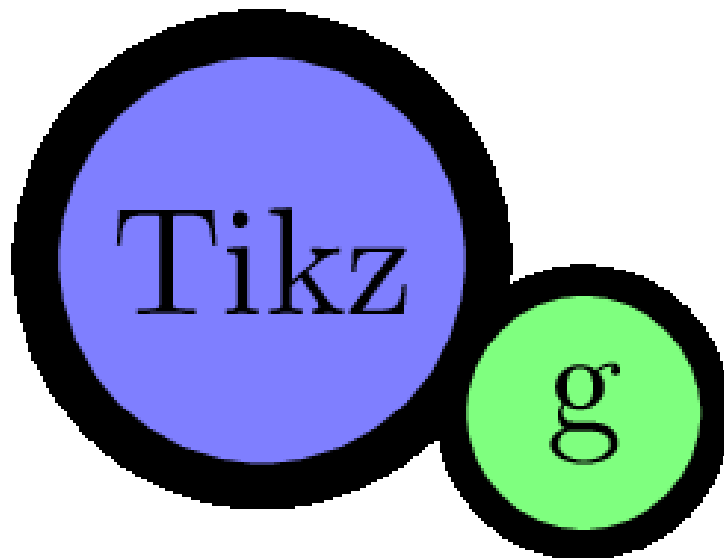


Table des matières

0.1	Avant propos	1
0.1.1	Qu'est ce que le logiciel TIKZG ?	1
0.1.2	Qu'offre le logiciel TIKZG ?	1
0.2	Installation du logiciel	3
0.3	Utilisation du logiciel TIKZG	3
0.3.1	Au démarrage	3
0.3.2	Barre de menu	4
0.3.3	Génération de graphe	6
0.3.4	Modification graphique des propriétés d'un nœud .	7

0.2 Installation du logiciel

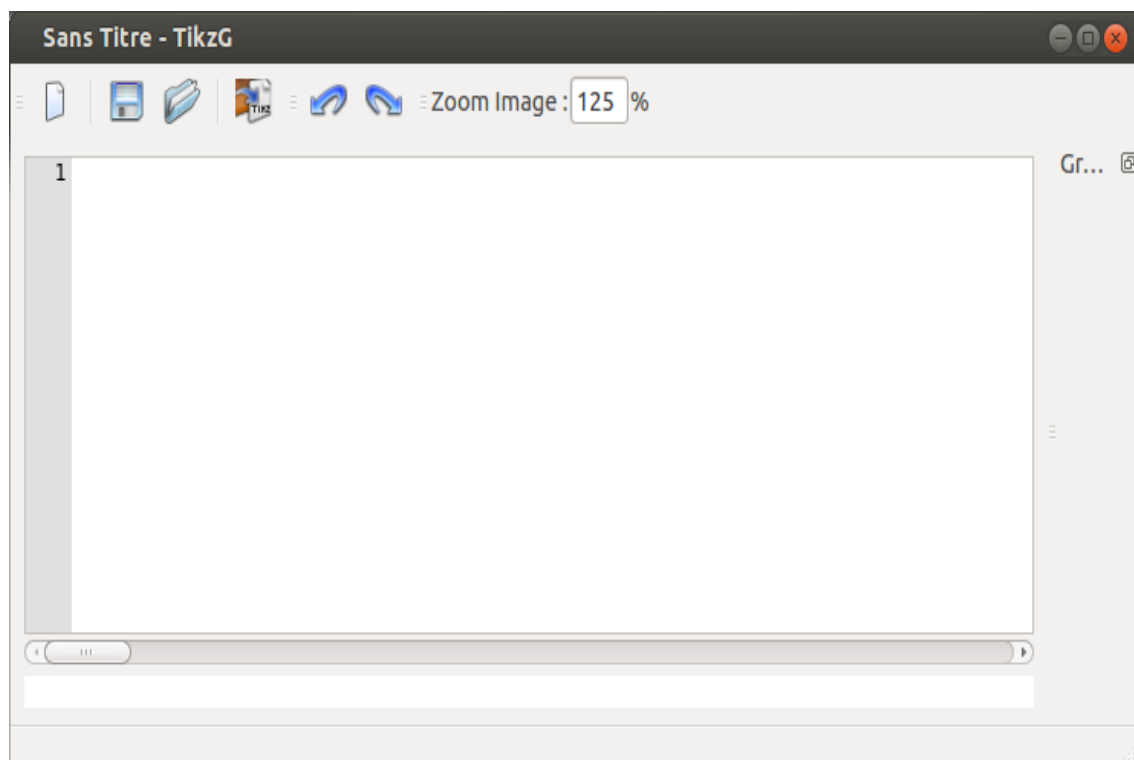
Conditions préalables : Votre ordinateur doit être équipé du système d'exploitation libre **Ubuntu**. Ensuite vous devrez récupérer TikzG qui est un logiciel libre.

Après récupération du logiciel, lancez le script d'installation du logiciel "script_install_all.sh". Une fois toutes les dépendances installées, ce qui peut prendre près d'une vingtaine de minutes, lancez le logiciel à partir de votre terminal avec la commande "perl tikzg.pl".

0.3 Utilisation du logiciel TIKZG

0.3.1 Au démarrage

Maintenant que vous avez installé votre logiciel, voici la fenêtre que vous devriez avoir au lancement de TIKZG.



Attention : le design de l'interface peut ne pas être le même, cela dépend de la version de Qt dont vous disposez. Sur l'image ci-dessus la version utilisée est Qt4.

0.3.2 Barre de menu

Fichier

Lorsqu'on clique sur **Fichier** dans la barre de menu, on a plusieurs fonctionnalités :

- Nouveau(raccourci ctrl+n) : permet de créer un fichier vierge dans l'éditeur de texte.
- Ouvrir(raccourci ctrl+o) : permet de récupérer un fichier Tikz et de l'ouvrir dans l'éditeur de texte et ainsi pouvoir visualiser le graphe généré.
- Enregistrer(raccourci ctrl+s) : permet d'enregistrer le code Tikz
- Enregistrer sous(raccourci ctrl+shift+s) : même principe que "Enregistrer" et permet en plus de choisir l'emplacement et le nom du fichier.
- Exporter : permet d'exporter le graphe obtenu sous forme d'image .png ou de fichier .pdf.
- Copier source figure (raccourci ctrl+shift+c) : permet de copier le code Tikz après lui avoir ajouté une ligne au début "`\begin{tikz}`" et une à la fin "`\end{tikz}`" afin d'obtenir un code Tikz complet et prêt à être utilisé.
- Imprimer code tikz : permet comme son intitulé l'indique d'imprimer le code source.
- Quitter (raccourci ctrl+q) : permet de quitter l'application

Editer

Lorsqu'on clique sur **Editer** dans la barre de menu, on a plusieurs fonctionnalités :

- Défaire(raccourci ctrl+z) : permet d'annuler la dernière saisie ajoutée au code.

- Refaire(raccourci ctrl+y) : permet de récupérer la dernière saisie annulée.
- Couper(raccourci ctrl+x) : permet de couper le texte sélectionné.
- Copier(raccourci ctrl+c) : permet de copier le texte sélectionné.
- Coller(raccourci ctrl+v) : permet de coller à l'endroit un texte préalablement copié.
- Supprimer : permet de supprimer le texte sélectionné.
- Tout sélectionner(raccourci ctrl+a) : permet de sélectionner tout le code.

Affichage

Lorsqu'on clique sur **Affichage** dans la barre de menu, on a plusieurs fonctionnalités :

- Affichage des numéros de ligne : permet d'afficher ou de cacher les numéros de ligne.
- Coloration syntaxique : permet d'activer ou de désactiver la coloration syntaxique.
- Générer image : permet de générer le graphe à partir du code tikz.

Aide

Lorsqu'on clique sur **Aide** dans la barre de menu on obtient comme seule fonctionnalité l'"A propos" qui nous fournit des renseignements sur l'application.

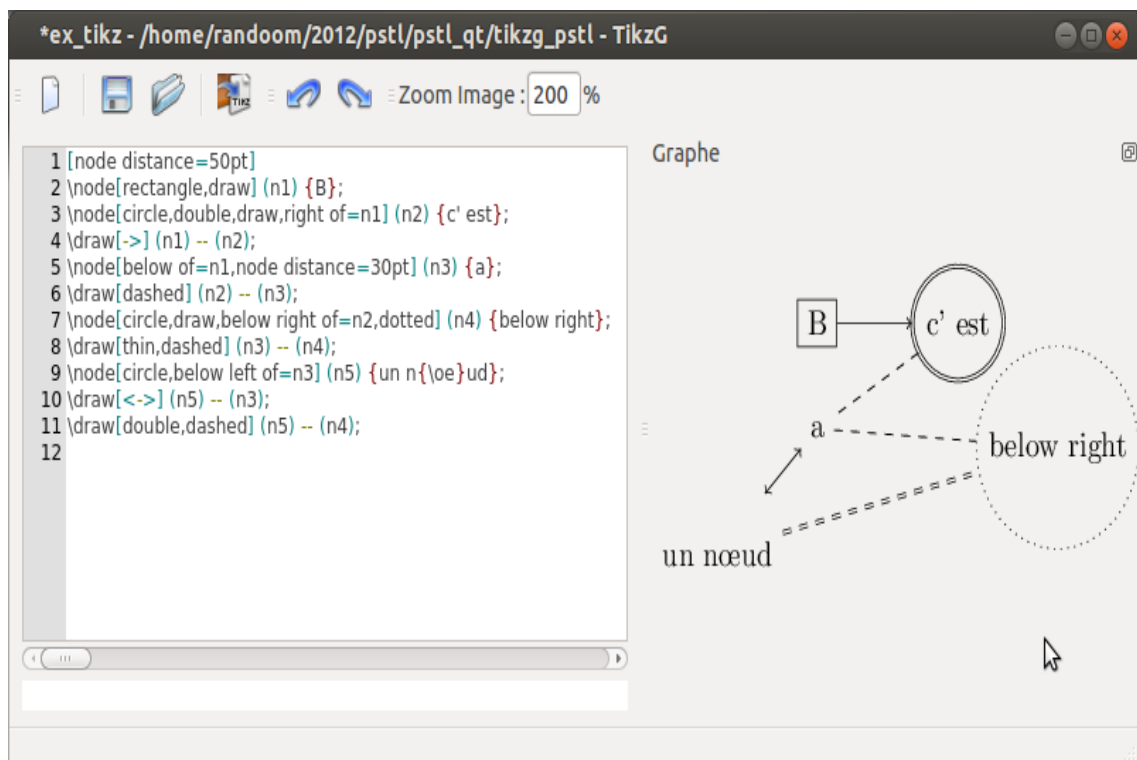
Zoom

Pour zoomer ou dézoomer le graphe, vous pourrez utiliser la molette de votre souris ou saisir une valeur de zoom dans la barre d'outil.

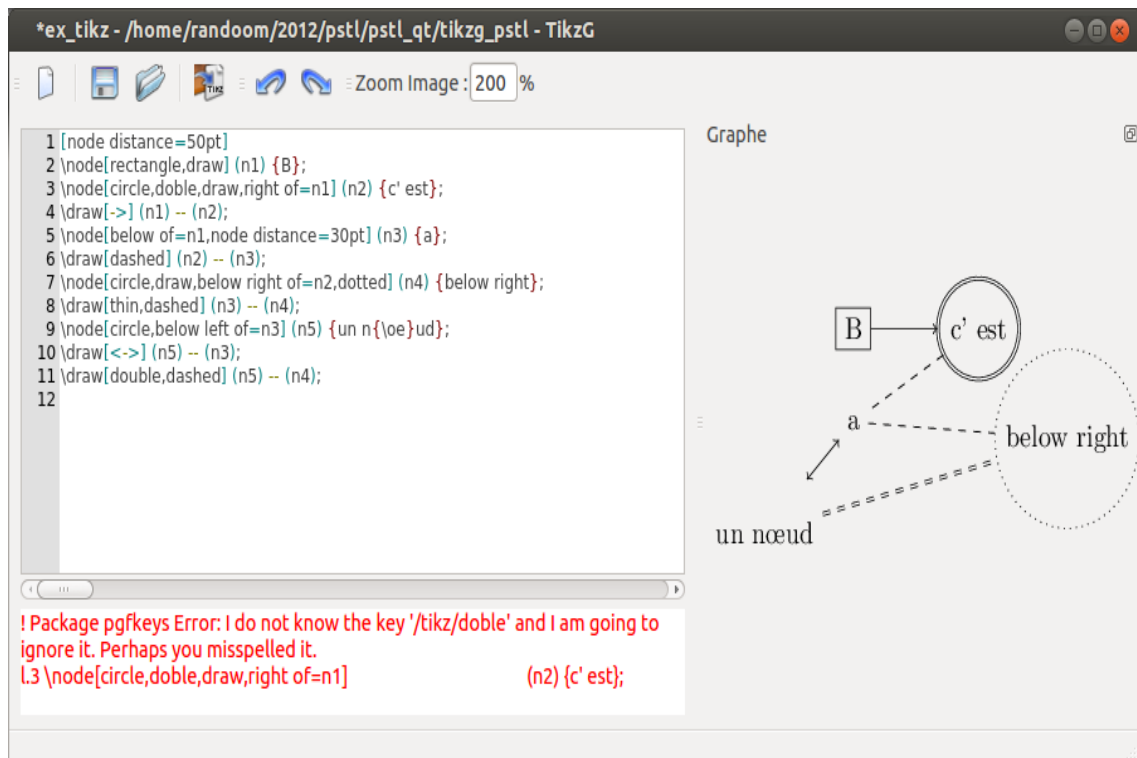
0.3.3 Génération de graphe

Le graphe est automatiquement généré à partir du code tikz au bout d'une seconde d'inactivité après modification de ce dernier. Voilà

un exemple de ce que vous pourrez obtenir avec le code saisi dans l'interface ci-dessous.

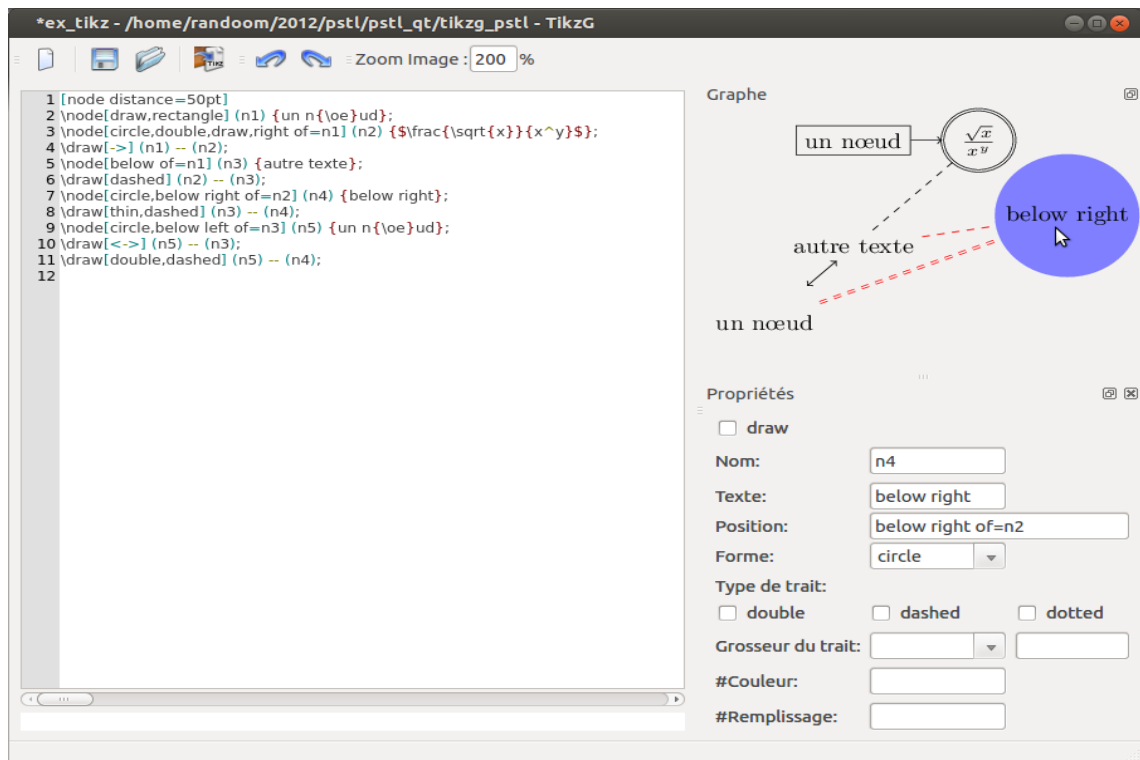


Et avec un code erroné vous verrez apparaître une fenêtre vous signalant vos erreurs et les lignes concernées comme ceci.



0.3.4 Modification des propriétés d'un nœud

Lorsque vous cliquez sur un nœud du graphe, le nœud sélectionné se colorie en bleu, ses nœuds relatifs en rose et les arêtes auxquelles il est relié en rouge. Une fenêtre contenant les propriétés du nœud en question s'affiche juste en-dessous du graphe. Vous devriez alors avoir une interface semblable à l'image ci-dessous.



Vous pourrez grâce à la fenêtre des propriétés choisir de dessiner(resp. cacher) le nœud sélectionné en cochant(resp. décochant) la case "draw". Vous pourrez aussi modifier le texte inscrit dans le nœud, le nom, la forme ou encore la position relative du nœud en modifiant les champs correspondants dans la fenêtre "Propriétés".