

Proposition de projet : TriComp

3 Octobre 2014

William AUFORT

Julien BENSMAIL
coordinateur

Agathe HERROU
chef de projet

Romain LABOLLE

Frédéric LANG

Maxime LESOURD

Laureline PINAULT

Léo STÉFANESCO

Résumé

Le but de ce projet est de fournir un “compilateur pour tricot” : un logiciel qui transforme des spécifications générales sur un tricot à réaliser (exemple : forme, couleurs, dimensions, types de points, ...) en des instructions à effectuer pour réaliser le tricot (exemple : faire 10 rangs en alternant mailles à l’endroit, mailles à l’envers, puis ...). L’idée serait même que le logiciel serve d’assistant au tricot en détaillant pour un non-expert du tricot (mais sachant tout de même tricoter!) les étapes les plus difficiles.

Ce projet sera réalisé dans le cadre d’une UE “projet intégré” de l’ENS de Lyon. L’équipe est composée de 7 étudiants en M1 d’informatique à l’ENS de Lyon. Le projet se déroulera entre octobre et mi-décembre.

Ce document vise à présenter notre proposition de projet. Il détaille la motivation derrière ce projet, l’objectif de ce projet, la manière dont nous comptons atteindre cet objectif (les différents composants du logiciel) ainsi que l’organisation (les différentes tâches, leur répartition au sein de l’équipe, le calendrier prévisionnel).

Table des matières

1	Introduction et objectifs	2
2	Produits concurrents	2
2.1	OpenKnit	2
2.2	DesignaKnit	2
2.3	KP	3
2.4	KnitML	3
2.5	Premières conclusions	3
3	Cahier des charges et applications	3
3.1	Fonctionnalités	3
3.1.1	Cinq différents niveaux de langages	3
3.1.2	Compilation	5
3.1.3	Réalisation logicielle : interface pour tricoteur	5
3.1.4	Diagramme des interactions	5
3.2	Aspects techniques	6
3.2.1	Portabilité	6
3.2.2	Langage de programmation	6
3.2.3	Licence	6
3.3	Public visé et retombées attendues	6
3.4	Améliorations possibles	7
4	Organisation et planification	7
4.1	L'équipe	7
4.2	Les différentes phases	7
4.3	Les différentes versions	8
4.4	Groupes de travail et tâches	8
4.5	Dépendances des tâches	10
4.6	Calendrier prévisionnel	10

1 Introduction et objectifs

Le tricot est une technique utilisée pour créer une étoffe à partir de fils. Ses origines remontent au X^{eme} siècle. Redevenu à la mode dans toutes les tranches d'âge, il a aussi connu un développement au travers de l'informatique. Il existe en effet quelques logiciels relatifs au tricot, mais ceux-ci visent un public plutôt expert, que ce soit en tricot ou en informatique (comment décrire un tricot à l'ordinateur?).

Le principe du tricot est de former des boucles de fil (appelées *mailles*), et de, à l'aide d'aiguilles, créer de nouvelles boucles imbriquées dans les précédentes, formant ainsi un nouveau *rang*. Chaque maille est reliée aux mailles qui lui sont adjacentes, permettant ainsi de créer une étoffe consistante. La combinaison des différentes manières de former les mailles et de les relier entre elles permet de définir une grande variété de *points*, et leur combinaison permet de créer une grande diversité de pièces.

Le projet TriComp visait au départ à concevoir un Compilateur pour Tricot (d'où son nom) afin de faciliter davantage l'interaction entre tricoteur et machine. Après un examen de l'état de l'art, nous avons davantage ciblé nos objectifs. Notre objectif principal est de fournir un logiciel disposant de fonctionnalités à la fois de création (de modèles), de visualisation et de conception (tricot), destiné au plus grand nombre (c'est-à-dire sans connaissances poussées en informatique, et pour des tricoteurs de tous niveaux).

La première des motivations est venue lors de la présentation du sujet. Ce sujet, à la fois troublant et surprenant, nous a interpellé et intéressé dès le début. Il s'agit d'un projet original, qui répond à un manque de logiciels simples d'utilisation pour tricoteurs de tous niveaux. De plus, la diversité des tâches à accomplir dans ce projet (aussi bien définitions "théoriques" (langage) que pratiques (compilateur, interface)) a permis à chaque membre de l'équipe de trouver sa place.

Ce document présente notre proposition de projet. Après avoir détaillé l'état de l'art des logiciels de tricot existants, nous exposerons notre cahier des charges et nos objectifs dans le détail. Enfin, nous détaillerons le déroulement du projet, à savoir les différents groupes de travail identifiés ainsi qu'un calendrier rassemblant les différentes tâches.

2 Produits concurrents

Nous présentons dans cette section les différents produits concurrents que l'on peut trouver, avec pour chacun une description du produit, ses avantages et ses inconvénients. Ainsi nous pourrions cibler davantage les fonctionnalités intéressantes à mettre en place et le travail déjà fourni pouvant nous être utile.

2.1 OpenKnit

OpenKnit¹ est une machine qui permet à ses utilisateurs de concevoir des vêtements en entrant leurs instructions dans un contrôleur Arduino (une carte programmable). Même si l'utilisateur peut créer ou télécharger des modèles, ce projet nécessite des connaissances en électronique assez importantes, et est plutôt orienté vers la fabrication du produit que sa conception (les instructions sont directement transmises à la machine OpenKnit). Il est donc plutôt destiné à un public qui souhaite faire fabriquer de simples créations et non pas aux tricoteurs (ce qui n'enlève rien au caractère innovant de ce projet).

Avantages : Fabrication possible directement à partir de l'ordinateur.

Inconvénients : Peu axé sur l'aspect tricoteur, production de qualité moindre...

2.2 DesignaKnit

DesignaKnit² est un logiciel permettant de créer des modèles de tricot destinés à être réalisés à la main ou avec une machine à tricoter. Destiné plutôt à un public confirmé, celui-ci est assez complet et

1. <http://openknit.org/>

2. <https://www.softbyte.co.uk/designaknit.htm>

propose quelques outils innovants, comme un convertisseur d'images vers un motif réalisable en tricot.

Avantages : Plutôt complet avec quelques outils intéressants.

Inconvénients : Uniquement de la conception.

2.3 KP

KP³ est un langage bas niveau développé en 2004 pour aider les tricoteurs à écrire des modèles de tricot. Il offre la possibilité de combiner ou d'ajuster des modèles pour en créer de nouveaux. Ce langage est accompagné d'un interpréteur qui génère des instructions elles aussi bas niveau. Bien qu'il fut abandonné avant d'être abouti, ce projet peut représenter une base pour l'élaboration de la phase plus théorique du projet.

Avantages : Une bonne base théorique pour formaliser le tricot.

Inconvénients : Inutilisable directement par le public.

2.4 KnitML

Le projet KnitML⁴ a également pour but de fournir un langage bas niveau au service des tricoteurs et développeurs de logiciels liés au tricot. Par exemple, le logiciel Knitter se base sur KnitML pour fournir une visualisation 3D du modèle passé en entrée. KnitML est davantage actualisé (la dernière version date de 2012) que KP et peut également nous apporter une précieuse aide.

Avantages : Base théorique intéressante, déjà utilisée par certains logiciels.

Inconvénients : ?

2.5 Premières conclusions

Des logiciels et des langages relatifs au tricot existent, chacun offrant des possibilités ciblées (fabrication, création, formalisation d'un langage). Cependant, aucun d'entre eux n'offre la possibilité de suivre un tricot en cours d'élaboration par l'utilisateur. De plus, les différents logiciels ont un public visé extrêmement restreint (trop complexes, investissement dans une machine, ...).

3 Cahier des charges et applications

3.1 Fonctionnalités

L'aboutissement de notre projet sera un logiciel destiné aux tricoteurs de tous niveaux, sans connaissances préalables en informatique. Ces utilisateurs pourront charger des modèles de vêtements prédéfinis (pulls, chaussette, écharpe...), les agrémenter (choix des points, de la couleur, ajouter des points plus exotiques...) et visualiser le résultat. Ensuite ils pourront générer les instructions correspondant à leur réalisation spécifique (à l'image de ce que l'on pourrait trouver dans un manuel de tricot) sous forme d'un tutoriel.

Pour remplir cet objectif, nous allons avoir besoin de plusieurs niveaux de langage afin de traiter les différents niveaux de représentation d'un tricot (d'une description "visuelle" aux instructions). Nous présentons ici ces langages ainsi que les composants du logiciel qu'ils vont les manipuler.

3.1.1 Cinq différents niveaux de langages

Notre logiciel va donc manipuler en tout cinq "langages", ou apparentés, de description des tricots. Ces cinq niveaux sont :

3. <http://www.cs.cmu.edu/~taey/pub/knit.pdf>

4. <http://www.knitml.com/blog/static.php?page=about-knitml>

- **Les spécifications utilisateur.** Au niveau de l'interface graphique, l'utilisateur pourra entrer les spécifications du tricot qu'il souhaite réaliser (forme, taille,...). En pratique, ce langage comprendra en particulier une énumération des objets à tricoter (pull, bonnet, gant...), des différents motifs choisis par l'utilisateur avec leur position sur le tricot, ainsi que certaines indications spatiales, comme la taille du tricot. Ainsi, une phrase dans ce langage consiste en une liste des propriétés du tricot souhaité.
- **Un langage descriptif.** Un des objectifs primordiaux de ce projet est de concevoir puis de formaliser un langage capable de décrire les tricots de manière générale. L'idée est que l'utilisateur qui veut tricoter un objet avec une forme très particulière puisse décrire son objet dans ce langage. A priori ce langage proposera à l'utilisateur d'assembler des formes de base (rectangles, trapèzes, cylindres,...) afin de décrire la forme de l'objet qu'il souhaite réaliser. Le passage des spécifications au langage descriptif sera effectué par l'interface graphique.
- **Une représentation intermédiaire.** Lors de la compilation du langage descriptif vers les instructions maille par maille, le compilateur utilisera une représentation intermédiaire, sous forme de graphe. Dans un tel graphe, les noeuds représenteront les mailles, tandis que les arêtes représenteront la manière dont elles sont reliées entre elles.
- **Langage bas niveau** Ce langage décrira les instructions à effectuer maille par maille afin de réaliser le tricot. Il sera très proche des instructions qui peuvent être données dans un manuel de tricot.
- **Instructions pour l'utilisateur.** Ce langage sera très proche du précédent. Il s'agit juste de rendre plus lisible par un humain la liste d'instructions définie par le langage bas niveau. Le traducteur s'occupera de passer de l'un à l'autre.

La figure 1 représente cette hiérarchie de langages.

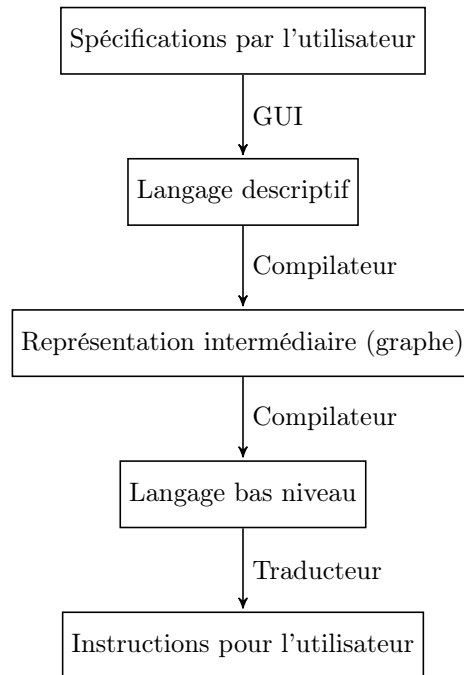


FIGURE 1 – Cinq niveaux de langages

3.1.2 Compilation

C'est l'intitulé du projet : construire un compilateur pour tricot. Le compilateur aura pour rôle de transformer le tricot que l'utilisateur souhaite créer en une interprétation dans un langage bas niveau (type KP), qui pourra ensuite être traduit en une succession d'instructions que celui-ci pourra réaliser. Le compilateur devra également être capable de détecter des impossibilités au niveau de la conception du modèle (deux point incompatibles qui se superposent au même endroit par exemple).

3.1.3 Réalisation logicielle : interface pour tricoteur

Notre réalisation logicielle a pour but de mettre le compilateur précédent au service des utilisateurs.

L'objectif ici est de concevoir un logiciel fournissant à ses utilisateurs essentiellement deux fonctionnalités :

- **La conception d'un tricot via une interface graphique.** Notre objectif ici est de mettre en place pour l'utilisateur des outils afin qu'il puisse définir son projet de tricot : choix d'un modèle (pull, écharpe...), des points utilisés (jersey, ...), de la couleur.
- **Le suivi personnalisé.** Une fois que l'utilisateur aura saisi son tricot, le logiciel appellera le compilateur afin de générer la liste des instructions qui devront être suivies pour réaliser le tricot. Ces instructions permettront un suivi du tricoteur par le logiciel : le tricoteur pourra à travers le logiciel suivre les différentes étapes, avec notamment des illustrations de la tâche à accomplir et l'observation du résultat à obtenir à chaque étape.

Un premier prototype d'interface a été conçu durant la première réunion. Cette interface comporterait trois zones principales :

- Une première zone contiendra la liste des éléments à choisir (points de tricot notamment) essentiellement sous la forme de boutons ;
- Une deuxième zone sera consacrée à la visualisation du modèle et du tutoriel généré ;
- Enfin, dans la dernière zone, l'utilisateur pourra faire défiler la suite d'instructions générée par le compilateur et ainsi suivre le bon déroulement de son tricot ; cette suite d'instruction pourra par ailleurs être exportée à l'extérieur du logiciel (pour par exemple être imprimée).

La figure 2 représente la disposition de ces différentes zones dans l'interface.

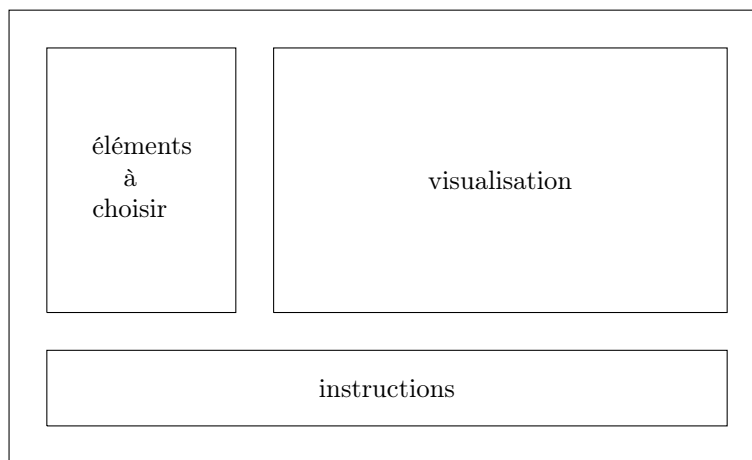


FIGURE 2 – Prototype de l'interface graphique

3.1.4 Diagramme des interactions

La figure 3 résume les interactions entre les différentes fonctionnalités.

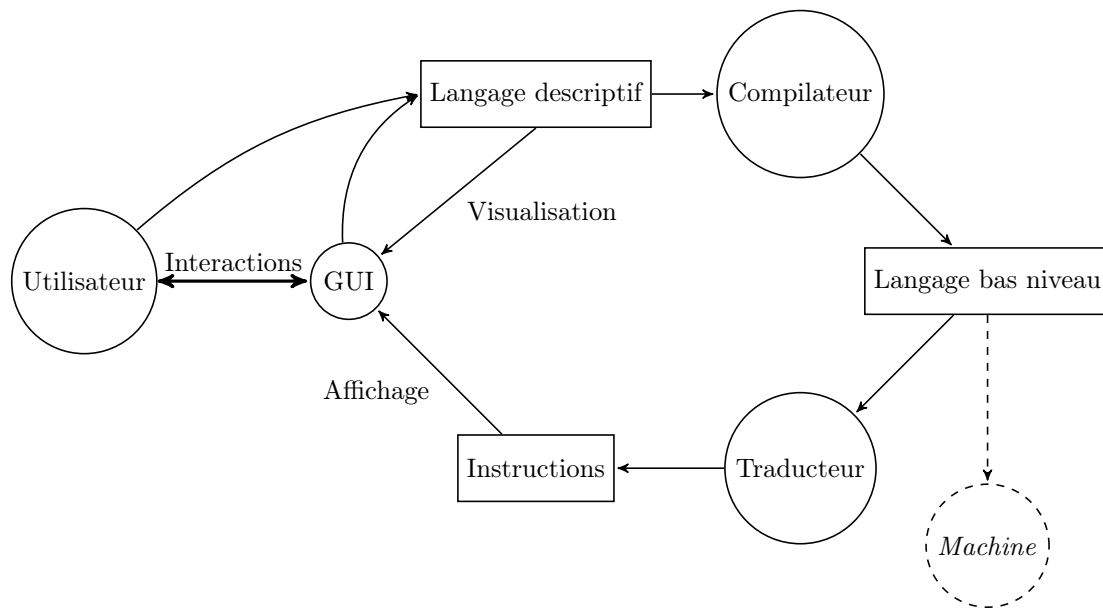


FIGURE 3 – Interaction entre les différents composants

3.2 Aspects techniques

3.2.1 Portabilité

TriComp est conçu comme un logiciel installable sur son ordinateur. Cela permet en effet une meilleure maîtrise de l’environnement et donc une meilleure stabilité. Cependant la portabilité est mise en avant et nous souhaitons développer un logiciel qui pourra s’installer à la fois sur Linux, Windows et MacOS. Pour cela nous utiliserons des outils multi-plateformes.

3.2.2 Langage de programmation

L’interface utilisateur sera développée grâce à un langage orienté objet. Nous avons choisi de programmer en C++ avec l’aide de la bibliothèque Qt (et éventuellement OpenGL si nous développons une visualisation 3D), ceci pour plusieurs raisons. Tout d’abord, Qt fournit beaucoup d’outils et une bonne documentation pour concevoir des interfaces graphiques. De plus, toutes les personnes qui seront engagées dans cette partie ont déjà une certaine familiarité avec ce langage.

Pour les modules ayant trait à la compilation et à la manipulation des langages en général, nous avons choisi OCaml, car en tant que langage fonctionnel, les tâches liées à la compilation (analyseur lexical et syntaxique...) seront plus simples à réaliser.

3.2.3 Licence

En ce qui concerne le choix de la licence, nous souhaitons produire un logiciel libre. Ainsi, nous nous sommes orientés vers la licence GPL V2.

3.3 Public visé et retombées attendues

Comme dit précédemment, notre projet vise tous les tricoteurs, experts ou non, recherchant un assistant dans la conception et la réalisation de leurs œuvres.

3.4 Améliorations possibles

Les améliorations possibles peuvent être séparées en deux catégories :

- La portée de chaque version : si le projet avance rapidement, nous pouvons espérer apporter plus de fonctionnalités (points de tricot, ...) aux versions successives.
- Des services inédits : on peut imaginer certains autres services, comme une visualisation des créations en 3D (typiquement sur un mannequin), une version multilingue du logiciel⁵, une option d'impression des instructions, un guide pour débutant (sous forme de manuel d'aide pour réaliser les points donnés dans les instructions), une application Android ou un traducteur vers une machine à tricoter.

Cependant, étant donné que l'objectif premier est de fournir un logiciel disposant d'un maximum de fonctionnalités de bases, il est fort possible qu'une grande partie ces améliorations (celles qui nécessitent beaucoup de temps de travail) ne soient pas réalisées durant le temps imparti pour la réalisation du projet.

4 Organisation et planification

4.1 L'équipe

L'équipe de notre projet TriComp comporte sept étudiants de M1 d'informatique de l'Ecole Normale Supérieure de Lyon ainsi qu'un coordinateur. Par ordre alphabétique : William AUFORT, Julien BENSMAIL (coordinateur), Agathe HERROU (chef de projet), Romain LABOLLE, Frédéric LANG, Maxime LESOURD, Laureline PINAULT et Léo STÉFANESCO.

Les tâches ont été réparties entre les différents membres de l'équipe (voir sections 4.4 et 4.6 pour plus de détails), qui peuvent travailler de manière (relativement) indépendante grâce à un dépôt git (<https://github.com/TriComp>). Des réunions sont prévues tous les mardis matin afin de faire le point sur les avancées de chacun et de discuter de la direction générale du projet.

4.2 Les différentes phases

Nous avons prévu de développer trois versions de notre logiciel. Chaque version sera basée sur la précédente et inclura de nouvelles fonctionnalités notamment au niveau de la variété et de la difficulté des points (de tricot) que notre logiciel prendra en compte.

Après une première phase de mise en route, les phases du projet correspondront aux différentes versions que nous avons prévu de réaliser :

La deuxième phase concerne la mise en place des briques de base de notre projet :

- Création d'une interface utilisateur (vide).
- Création et lancement du site web.
- Définition des langages. Le langage descriptif sera par la suite étendu, mais cette première phase permettra de définir la structure et les premières commandes de ce langage.

Une fois cette phase réalisée, les phases suivantes seront similaires et consisteront à fournir une version améliorée des différentes fonctionnalités. Pour chaque nouvelle version, différents groupes de travail (concernant respectivement le langage, le compilateur, le module de vérification ou l'interface graphique) seront développés en parallèle. Chacune de ces phases sera conclue par une intégration des nouveautés mises en place dans chacun des groupes de travail, ainsi que d'une série de tests afin de vérifier que l'intégration est réussie et que le logiciel répond aux attentes que nous nous sommes fixées (par exemple, nous pourrions le soumettre à des tricoteurs, et pas seulement vérifier que toutes les fonctionnalités sont opérationnelles).

5. en français et en anglais notamment...

4.3 Les différentes versions

La première version du langage de description décrirait une forme très simplifiée du tricot, ne faisant appel qu'aux points à l'endroit et à l'envers, c'est à dire à la base du tricot. Cette version ne devrait donc pas donner lieu à des motifs impossibles.

La deuxième version introduirait des points un peu plus difficiles à manipuler ; nous prévoyons donc qu'elle puisse manier des duplications et fusions de mailles (qui sont notamment utilisées pour créer des trous dans un ouvrage).

Enfin, la troisième version s'attaquerait à l'opération la plus complexe que nous avons prévu de manipuler : les tresses, où plusieurs portions de l'ouvrage peuvent se croiser, introduisant ainsi des difficultés supplémentaires dans le suivi du parcours du fil.

Au niveau des fonctionnalités, la visualisation des instructions sous forme de tutoriel ne sera intégrée au logiciel qu'à partir de la version 2.

4.4 Groupes de travail et tâches

Nous avons divisé le travail à effectuer en six différents groupes de travail, correspondant globalement aux différents composants du logiciel. Ces groupes de travail sont eux-mêmes subdivisés en tâches.

Ci-dessous le détail et la répartition humaine des tâches (la personne en gras est responsable du bon déroulement de la tâche) :

- 1. Langage bas niveau.** Ce groupe de travail est destiné à concevoir les instructions du langage bas niveau et ce qui tourne autour. Les tâches de ce groupe de travail sont les suivantes :
 - Définition du langage : Consiste à définir les instructions du langage.
Production : Une liste d'instructions décrivant le langage bas niveau.
Laureline
 - Module de traduction : Consiste à implémenter un traducteur qui traduira les instructions du langage bas niveau en des instructions en langage naturel, type manuel de tricot, compréhensible pour l'utilisateur.
Production : Un module traduisant le langage bas niveau en instructions de tricot compréhensibles pour l'utilisateur.
Laureline
- 2. Vérification.** A partir de la version 3, le logiciel permettra à l'utilisateur de définir des motifs par combinaisons de points. Or si ces motifs sont définis sans faire attention, il peut y avoir des conflits. Le module de vérification sert à repérer ces conflits lors de la compilation ainsi qu'à indiquer à l'utilisateur où et pourquoi ils ont lieu. Les tâches de ce groupe de travail sont les suivantes :
 - Définition des contraintes : Consiste à définir les conflits qui peuvent apparaître.
Production : Une liste de contraintes décrivant les conflits possibles.
Laureline, Fred, Agathe
 - Implémentation des contraintes : Consiste à implémenter un module de vérification qui se lancera avec la compilation pour rattraper les erreurs survenant lors de la compilation.
Production : Un module rattrapant les erreurs liées aux conflits lors de la compilation.
Laureline
- 3. Compilateur.** C'est le coeur du projet : le module qui va traduire le langage décrivant le tricot en instructions bas niveau, permettant de réaliser le tricot maille par maille. Ce groupe de travail est subdivisé en trois tâches semblables, une pour chaque version :
 - Définition de la représentation intermédiaire : Consiste à définir la représentation intermédiaire avec laquelle va travailler le compilateur.
Production : Une description de la représentation et une implémentation de celle-ci.
Agathe, Laureline

- Compilation (ième version) A chaque version, le langage descriptif sera étendu, d'où la nécessité de mettre à jour le compilateur pour qu'il prenne en compte les nouveaux éléments du langage. Cette tâche consiste à implémenter un compilateur transformant la ième version du langage descriptif en instructions bas niveau.

Production : Un module compilant le langage descriptif vers des instructions bas niveau.

Léo, Laureline, Maxime

4. Interface graphique. L'interface graphique sera le lien entre l'utilisateur et le logiciel. Elle doit permettre à l'utilisateur un accès ergonomique à toutes les fonctionnalités du logiciel. Elle va également permettre à l'utilisateur de visualiser ce qu'il spécifie et va lui servir d'assistant pour tricoter à partir des instructions fournies. Les tâches de ce groupe de travail sont les suivantes :

- Mise en place de l'interface : Créer l'interface vide, la coquille. Commencer à travailler sur la visualisation.

Production : Une interface graphique, pas encore reliée au logiciel

William, Romain

- Visualisation : Implémenter un module permettant, à partir d'une description en langage descriptif, visualiser l'objet décrit.

Production : Un module de visualisation pour le langage descriptif

William, Léo, Romain, Fred

- Traduction des spécifications : Développer dans l'interface un moyen ergonomique permettant à l'utilisateur d'entrer ses spécifications, lesquelles seront traduites en langage descriptif.

Production : Une interface graphique prenant en compte les nouveautés apparues dans la nouvelle version du langage descriptif, permettant à l'utilisateur de décrire de manière simple et intuitive ce qu'il veut tricoter.

Fred, Agathe

- Affichage des instructions : Afficher les instructions utilisateur sous forme de tutoriel dans l'interface graphique.

Production : Intégrer l'affichage des instructions dans l'interface graphique.

Laureline, William

5. Langage Descriptif. Ce groupe de travail est destiné à définir le langage haut niveau permettant de décrire les objets à tricoter et à modéliser les différents points.

- Définition du langage : Définir un langage permettant de décrire le tricot, avec les fonctionnalités et la variation de points disponibles définis dans les spécifications de la version courante.

Production : Un langage descriptif, étendu à chaque version.

Maxime, Léo, Agathe

6. Communication. Ce groupe de travail inclut tout ce qui concerne la communication autour du projet, que ce soit au niveau large public (site web), au niveau utilisateur (readme et manuel) ou au niveau de l'équipe (comptes-rendus de projet (proposition, rapport d'activité mi-parcours et rapport final), initiation au tricot, mise au point chaque semaine).

- Proposition de projet : Rédiger la proposition.

Production : Proposition

Laureline, William

- Rapport d'activité mi-parcours : Rédiger le rapport d'activité mi-parcours. Celui-ci contiendra tout le travail réalisé jusque là, et notamment le descriptif de la version 1 qui devrait être fonctionnelle d'ici là.

Production : Rapport d'activité mi-parcours

William, tout le reste de l'équipe

- Rapport final et présentation : Rédiger le rapport de fin de projet et préparer la présentation du projet.

Production : Rapport final et diapositives de la présentation.

Agathe, tout le reste de l'équipe

- Site web : Créer un site web pour le projet.
Production : Site web
Romain, Fred
- Readme, Manuel et Site web : Mettre à jour le readme, le manuel et le site web pour qu'ils soient cohérents avec la version en cours du logiciel.
Production : Readme, manuel et site web
Romain, Fred

4.5 Dépendances des tâches

Au niveau des tâches tournant autour du langage bas niveau, la tâche de transformation des instructions utilisateur en tutoriel affiché dans l'interface graphique doit se faire après la tâche de traduction du langage bas niveau en instructions utilisateur, qui elle-même doit se faire après la tâche de définition du langage bas niveau.

Au niveau des tâche de compilation, pour chaque version, les tâches de compilation, de visualisation, de traduction des spécification et de vérification (quand elle est nécessaire) peuvent se faire en parallèle, mais seulement après la définition du langage descriptif.

Enfin, les version se font successivement dans le temps, car elles servent de base les une aux autres.

4.6 Calendrier prévisionnel

Globalement, nous comptons un délai de trois semaines pour chaque version (sauf la première qui sera plus longue à mettre en route). L'agencement précis des tâches dans le temps a été fait en prenant en compte les dépendances de chacune des tâches.

Nous décrivons dans cette section la répartition temporelle des tâches précédentes avec un planning (voir ci-dessous). Il s'agit d'un planning sur un document collaboratif, qui sera mis à jour au cours du projet.

Pour la plupart des tâches, deux semaines sont prévues. Le principe est que si une tâche prend plus de temps que prévu, retardant de la même toutes les autres dépendant d'elle, nous essayerons d'effectuer celles-ci en une semaine. De plus la dernière semaine étant uniquement consacrée à des tâches de rédaction, elle pourrait servir à rattraper d'éventuels retards.

WORKPACKAGES	CALENDRIER										
		Définition du langage	Lg BN => instructions utilisateur								
Langage bas niveau		Laureline	Laureline								
		Laureline	Laureline								
								Définition des contraintes	Implémentation		
Vérification								Laureline	Laureline	I	
								Laureline, Fred, Agathe	Laureline, Agathe	N	
		Définition de la représentation intermédiaire	Compilateur v.1	I	Compilateur v.2		Compilateur v.3			TE	
Compilateur		Agathe	Léo	N	Léo	I	Léo			G	
		Agathe, Laureline	Léo, Laureline, Maxime	T	Léo, Laureline, Maxime	N	Léo, Laureline, Maxime			R	
		Interface v.0	Visualisation v.1	E	Visualisation v.2	TE	Visualisation v.3			A	
		William	William	G	William	G	William			T	
		William, Romain	William, Léo, Romain, Fred	R	William, Léo, Romain, Fred	R	William, Léo, Romain, Fred			I	
			Trad. Spécifications v.1	A	Trad. Spécifications v.2	A	Trad Spécifications v.3			O	
Interface graphique			Fred	T	Fred	T	Fred			N	
			Fred, Agathe	ION	Fred, Agathe	I	Fred, Agathe				
					Instructions utilisateurs => manuel	O					
					Laureline	N					
					Laureline, William						
		Définition du langage v.1	Définition du langage v.2		Définition du langage v.3						
Langage descriptif		Maxime	Maxime		Maxime						
		Léo, Maxime, Agathe	Léo, Maxime, Agathe		Léo, Maxime, Agathe						
			Site		Readme + Manuel + Site v.1		Readme + Manuel + Site v.2			Readme + Manuel + Site v.3	
			Romain		Romain		Romain			Romain	
Communication			Fred, Romain		Fred, Romain		Fred, Romain			Fred, Romain	
	Proposal				Mid-term Rapport					Final Rapport + Presentation	
	Laureline				William					Agathe	
	Laureline, William				All					All	
		29/09	13/10	27/10	03/11	04/11	17/11	24/11	01/12	08/12	15/12
					v1 fonctionnelle			v2 fonctionnelle			v3 fonctionnelle