

# Rapport d'activité mi-parcours : TriComp

4 Novembre 2014

William AUFORT

Julien BENSMAIL  
coordinateur

Agathe HERROU  
chef de projet

Romain LABOLLE

Frédéric LANG

Maxime LESOURD

Laureline PINAULT

Léo STÉFANESCO

## Résumé

Ce document présente le rapport d'activité mi-parcours de notre projet TriComp. Y sont détaillés le travail fourni jusqu'à présent dans les différents groupes de travail ainsi que les diverses modifications qui ont été effectuées par rapport à la proposition de projet.

# Table des matières

<b>1</b>	<b>Travail fourni</b>	<b>2</b>
1.1	Définition des différents langages . . . . .	2
1.1.1	Instructions utilisateurs . . . . .	2
1.1.2	Langage descriptif . . . . .	3
1.2	Interface graphique . . . . .	4
1.2.1	Squelette de l'interface / Aspect . . . . .	4
1.2.2	Affichage des tricots . . . . .	4
1.3	Site Web . . . . .	5
<b>2</b>	<b>Changement par rapport à la proposition</b>	<b>5</b>
2.1	Objectifs . . . . .	5
2.1.1	Représentation intermédiaire . . . . .	5
2.2	Calendrier . . . . .	6
2.3	Précisions diverses . . . . .	6

# 1 Travail fourni

Nous exposons dans cette partie le travail fourni jusqu'à présent dans les différents groupes de travail.

## 1.1 Définition des différents langages

L'étape cruciale de définition des langages a été effectuée. Nous exposons ici avec précision ces différents langages.

### 1.1.1 Instructions utilisateurs

Une première version du langage des instructions utilisateurs a été établie, avec des tests sur des exemples. Nous avons maintenant des outils suffisants pour expliquer comment tricoter un modèle simple, comme par exemple un tricot rectangulaire avec un motif régulier (c'est le cas pour une écharpe).

Concrètement, une description bas-niveau d'un tricot, quasiment identique aux instructions utilisateurs, mais en langage machine, a été mise en place, le tricot est ici vu comme un type OCaml :

```
type stitch      = Endroit | Envers | Torse | Vide
type row         = stitch list * int
type parallelogram = row list * int
type piece       = parallelogram list
```

Une maille (stitch) est décrite par son type : endroit, envers ou torse. Le type Vide est utilisé pour décrire les premières mailles montées (et sera également utilisé par la suite pour décrire des augmentations). Une ligne (row) est décrite par le motif minimal qui la décrit (sous forme d'une liste de mailles) et par un entier indiquant le nombre de fois que ce motif doit être répété. De la même manière un rectangle tricoté (parallelogram) est décrit par le motif minimal qui la décrit (sous forme d'une liste de lignes) et par un entier indiquant le nombre de fois que ce motif doit être répété. Enfin une pièce de tricot (piece) est décrite par une liste de rectangles tricotés.

Depuis cette description, on peut obtenir les instructions utilisateurs en français pour réaliser le tricot. Pour cela on utilise un traducteur que nous avons implémenté.

L'exemple suivant donne une liste d'instructions que l'on peut obtenir avec le traducteur :

**Exemple 1.** *Suivez les instructions suivantes pour obtenir votre tricot :*

- Rang 0 : Tricotez un point vide, et répétez 12 fois ce motif.
- Rang 1 : Tricotez un point endroit puis un point envers, et répétez 12 fois ce motif.
- Rang 2 : Tricotez un point envers puis un point endroit puis un point envers, et répétez 8 fois ce motif.
- Rang 3 : Tricotez un point envers puis un point endroit, et répétez 12 fois ce motif.
- Rang 4 : Tricotez un point endroit puis un point envers, puis un point envers et répétez 8 fois ce motif.
- Tricotez au total 10 fois ces 4 derniers rangs.

Par ailleurs nous avons réfléchi à la forme que nous souhaitons donner aux instructions utilisateurs pour des tricots plus complexes, des tricots nécessitant la couture de plusieurs pièces tricotées indépendamment par exemple. Le format de sortie souhaité ressemblerait à ça :

#### Titre du tricot

*On présente d'abord potentiellement quelques informations préliminaires : matériel requis, auteur du modèle...*

### Présentation du patron du tricot

*Liste des différentes parties à tricoter, dans l'ordre où elles doivent être tricotées.*

#### 1eme pièce à tricoter

*Instructions pour tricoter la 1eme pièce*

Un exemple :

- aiguilles et laine à utiliser
- lancer le tricot avec 21 boucles sur une première aiguille (ligne 0)
- instructions :

\* 1ere ligne : un point endroit, un point envers, un point endroit, puis répéter ce motif 6 fois (jusqu'au bout de la ligne)

\* 2e ligne : un point envers, un point envers, un point endroit, puis répéter ce motif 6 fois (jusqu'au bout de la ligne)

\* Répéter ces 2 lignes 10 fois pour obtenir 22 lignes

\* 23e ligne : un point envers, un point endroit, puis répéter ce motif 4 fois. Ensuite faire une diminution, et encore répéter le motif 5 fois.

#### Keme méta-instruction

*Instructions d'assemblages*

Un exemple :

- Prendre la partie I.
- Coudre son bord gauche avec son bord droit.

### 1.1.2 Langage descriptif

Le langage descriptif permet d'interfacer le traducteur et l'interface graphique. Les contraintes étaient d'avoir un format de description facilement manipulable à travers l'interface graphique tout en gardant la structure d'un tricot. La solution adoptée est de voir un tricot comme un assemblage d'éléments qui sont des structures arborescentes de trapèzes tricotés. Ces trapèzes sont tricotés selon divers motifs, pour le moment en point endroit ou envers. Plus précisément, le langage décrit les relations entre les trapèzes se touchant via leurs bases. Outre le cas où deux trapèzes partagent la même base, la base d'un trapèze peut être adjacente à plusieurs trapèzes : on décrit alors leur séparation via "split" et la réunion de deux trapèzes via "link". Les bases correspondant au début et à la fin du tricot sont indiquées par les mots clés "start" et "stop".

**Exemple 2.** *Un code du type de celui-ci décrit un poncho, illustré par la figure 1.*

```
piece my_piece := start size
                || trapezoid (parametres)
                || split { trapezoid (...)
                        || link left next
                        }{ trapezoid (...)
                        || link right next
                        }
                }

piece next := start size
            || trapezoid (...)
            || stop
```

La syntaxe a été définie et le parser utilisé par le traducteur a été implémenté. Conserver les trapèzes est selon nous intéressant lorsque l'utilisateur éditera son tricot. Chaque modification correspondra à une séparation des trapèzes, chose qui est déjà gérée par notre langage.

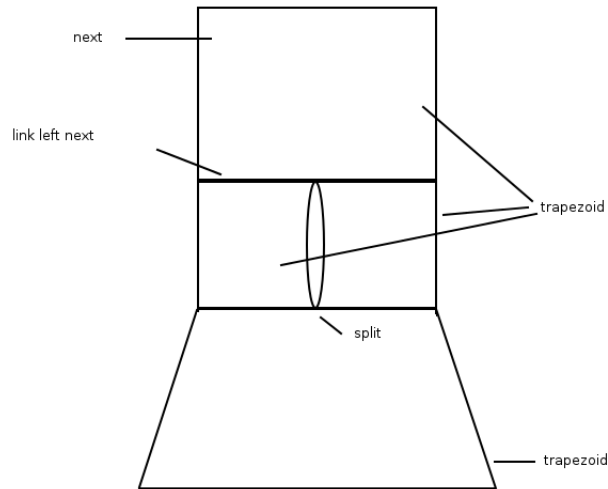


FIGURE 1 – Le poncho généré par le code précédent

## 1.2 Interface graphique

Nous détaillons dans cette section les fonctionnalités implémentées jusqu'à présent au niveau de l'interface graphique.

### 1.2.1 Squelette de l'interface / Aspect

La première étape du développement de l'interface a été la mise en place de son squelette. Plus précisément, nous avons défini l'aspect général de l'interface ainsi que les différents outils que nous souhaitons mettre à disposition. Ces outils sont représentés à l'aide de boutons ou d'options encore inactives pour la plupart. On distingue notamment :

- Les options relatives au logiciel TriComp (choisir un point, faire une tresse...) Ceux-ci peuvent être sélectionnés grâce à des boutons dont l'aspect n'est pas encore fixé, car leur implémentation se fera au fur et à mesure et en fonction de l'avancement global du projet.
- Les options que l'on trouve dans tout logiciel (ouvrir, sauvegarder, quitter, ...). Ces options sont fonctionnelles (ou bientôt fonctionnelles) car les formats de données pour la sauvegarde ont été définis. Nous travaillons directement sur le format de fichier associé au langage descriptif. Ce type de fichier portera l'extension `.tricot`.

La figure 2 présente l'interface utilisateur du logiciel. Celle-ci se décompose en trois parties :

1. un panneau qui contiendra les différents outils de tricot mis à la disposition de l'utilisateur. Ces outils seront sous le forme de boutons ;
2. une fenêtre d'affichage du tricot ;
3. une fenêtre qui contiendra la liste des instructions générées par l'utilisateur.

### 1.2.2 Affichage des tricots

L'affichage des tricots en lui-même est en cours d'implémentation. Il est prévu de les afficher sous forme de patron, avec éventuellement la possibilité d'afficher la progression d'un tricot en cours de

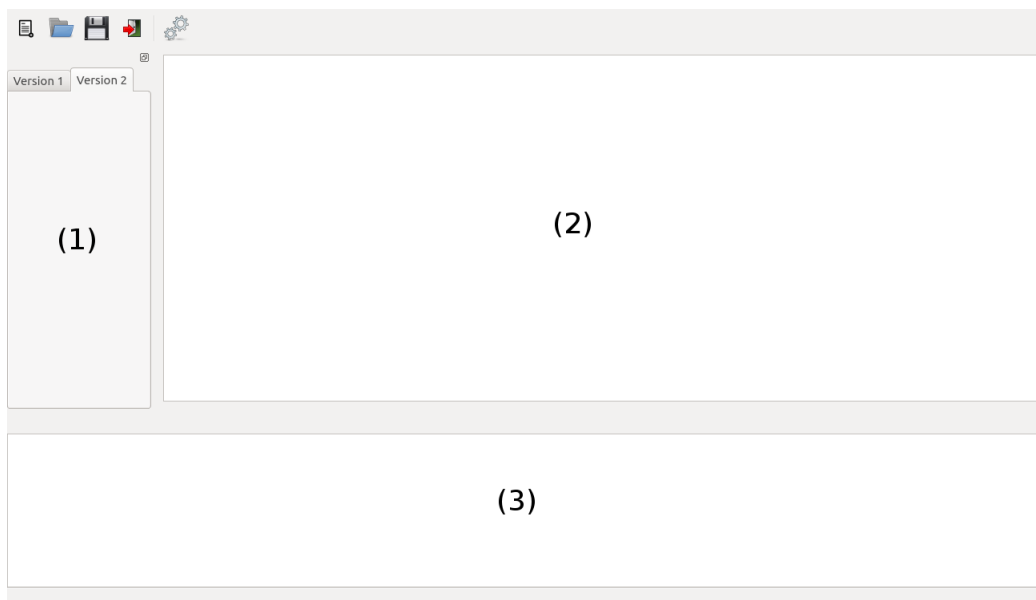


FIGURE 2 – L’aspect général de la fenêtre principale, avec les trois parties principales mises en évidence

réalisation. Nous travaillons avec des classes de Qt qui permettent la gestion et l’affichage d’objets 2D (`QtGraphicsScene` et `QtGraphicsItem` par exemple).

Nous avons également besoin d’un parser pour permettre l’utilisation du fichier `.tricot` par l’interface graphique. Celui-ci est sensiblement identique à celui implémenté en Ocaml et est en cours de finition (notamment pour l’incorporation dans le projet Qt).

### 1.3 Site Web

Un site web a été déployé à l’adresse <http://tricompile.github.io>, il est hébergé sur Github (utilisé comme serveur git du projet). Le site est basé sur Jekyll, un CMS en Ruby spécialisé dans les blogs et qui a la particularité de ne pas utiliser de base de données, avec l’avantage que Github gère Jekyll automatiquement. Le site est notamment une vitrine pour le projet : il en contient une présentation rapide, avec des liens vers le code source sur Github. A terme, le site sera aussi un support pour télécharger et installer le logiciel TriComp.

De plus, le site contient pour l’instant un article sur ce qui existe déjà sur Internet autour du tricot (notamment certains projets proches de TriComp qui ont été détaillés dans la proposition). D’autres articles devrait s’y rajouter, ainsi qu’une page pour aiguiller rapidement l’internaute anglophone.

## 2 Changement par rapport à la proposition

### 2.1 Objectifs

#### 2.1.1 Représentation intermédiaire

Il avait initialement été prévu de développer une représentation intermédiaire, où le tissu serait représenté sous la forme d’un graphe, dans lequel les sommets correspondraient aux mailles et les arêtes à la manière dont les mailles interagissent. Cette représentation avait pour but primaire d’aider à la détection de configurations impossibles. Cependant, après avoir approfondi les langages de bas et haut niveaux, nous nous sommes rendus compte que ce formalisme n’était pas intrinsèquement nécessaire.

Nous avons donc abandonné l'utilisation globale de cette formalisation, pour ne se concentrer que sur l'étude de motifs impossibles ponctuelle et non systématique.

## 2.2 Calendrier

**Interface graphique** Nous avons prévu d'avoir une interface graphique fonctionnelle dès lors que la première version du langage aurait été définie et compilable. Cependant, elle s'est révélée plus difficile à mettre en place que ce que nous avions imaginé, l'intégration du compilateur (lecture de fichiers `.tricot` (haut niveau) et transformation en instructions utilisateur (bas niveau). La deadline que nous nous étions fixée pour la version 1 du langage sera donc repoussée, et devient notre priorité.

**Versions supérieures du langage** La définition des versions 2 et 3 du langage n'a pas été faite formellement, mais les différences avec la version 1 se réduisant à l'ajout de nouvelles opérations au langage (respectivement diminutions/augmentations et croisements d'éléments de tricot), une telle définition n'était pas forcément nécessaire en dehors de l'implémentation. Cependant, les problèmes que pourraient introduire ses versions ont été étudiés, notamment la question de la répartition des diminutions sur un trapèze afin que sa pente soit régulière, ou les incompatibilités que pourraient poser l'introduction des tresses.

## 2.3 Précisions diverses

## Conclusion