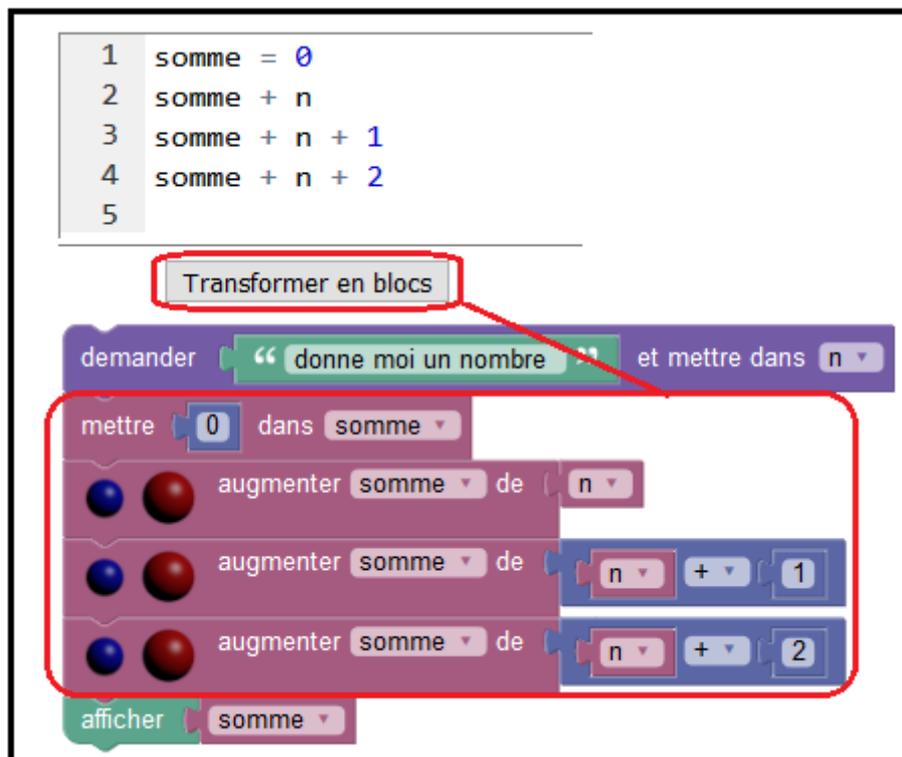


Annexe : transformation d'instructions textuelles en blocs

<http://revue.sesamath.net/spip.php?article1057>
patrick.raffinat@univ-pau.fr

A) Introduction

Dans l'article, j'ai présenté quelques commandes textuelles pouvant faire gagner du temps dans l'écriture d'un programme Blockly au collège :



Je montre dans cette annexe qu'il est possible d'aller plus loin dans la transformation d'instructions textuelles (algorithmiques ou Python) en blocs. Ces compléments ne sont donnés qu'à titre indicatif : non seulement le développement n'est pas terminé, mais je dois encore chercher à mieux cerner les apports pédagogiques que peuvent avoir ces nouvelles fonctionnalités au collège ou au lycée.

La programmation visuelle ayant été inventée pour s'affranchir des contraintes syntaxiques de la programmation textuelle, il peut paraître paradoxal de définir un mini-langage algorithmique pour créer des blocs. Mais il me semble par exemple que l'utilisation de boucles « répéter » dans un algorithme pour faire des figures avec une tortue pourrait intéresser des enseignants de collège.

Au lycée, cette nouvelle possibilité pourrait être utilisée pour transformer un algorithme en un programme Python : le programme Blockly engendré ne serait alors qu'un intermédiaire, à partir duquel les élèves pourraient obtenir un programme Python puisque SofusPy a un outil de traduction automatique de Blockly vers Python.

B) Exemple

Dans cet exemple qui dessine un triangle équilatéral, puis le fait pivoter de 120 et de 240 degrés, j'ai artificiellement introduit une instruction conditionnelle dont le seul but est d'illustrer les capacités actuelles de l'outil :

<pre># Version 1 : purement algorithmique n = 2+1 si n = 3 : taille = 50 + 5 + 15 taille + 2 * 5 pour k entre 1 et 3 : droite 120 repeter n : avancer taille gauche 360/n afficher taille afficher n # Version 2 : mélange algo + Python n=2+1 if (n==3) : taille = 50 + 5 + 15 taille + 2 * 5 pour k entre 1 et 3 : right(120) repeter n : forward(taille) left(360/n) print(taille) print(n)</pre>	
---	--

Le « : » terminant le « si », le « pour » et le « répéter » n'est pas nécessaire pour que ce pseudo-code soit correctement transformé en blocs, mais il permet à l'éditeur Python de SofusPy d'ajouter automatiquement une tabulation sur la ligne suivante, tabulation qui elle est indispensable : par exemple, si l'instruction « taille = 50+5+15 » n'avait pas de tabulation, elle serait rattachée après l'instruction conditionnelle et non à l'intérieur.

Voici la traduction en Python du programme Blockly, puis son exécution :

<pre>from turtle import * n = 2 + 1 if n == 3: taille = 50 + (5 + 15) taille = taille + 2 * 5 for k in range(1,4): right(120) for count in range(int(n)) forward(taille) left((360 / n)) print(taille) print(n)</pre>	
---	--