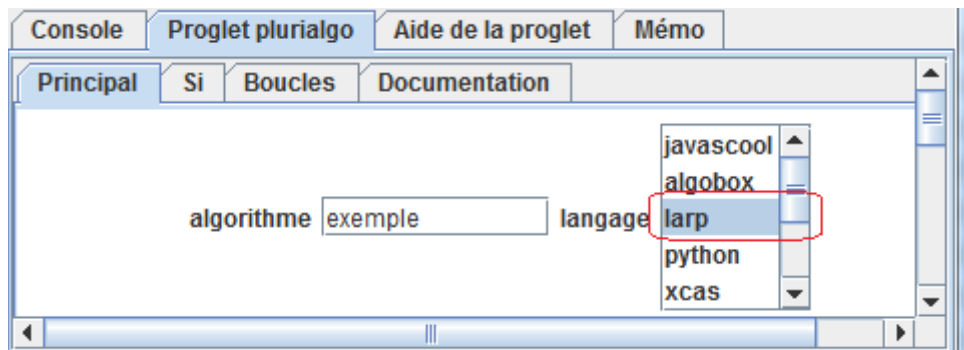


Problème de simulation avec PluriAlgo

Ce document détaille le problème de simulation abordé dans l'article principal : estimer la probabilité de l'évènement « au moins 60 piles sur 100 lancers ». Pour le résoudre, nous allons procéder en deux étapes :

1. en comptant le nombre de piles lors d'une simulation de 100 lancers d'une pièce.
2. en répétant la simulation de 100 lancers, afin de compter le nombre de fois où il y aura au moins 60 piles.

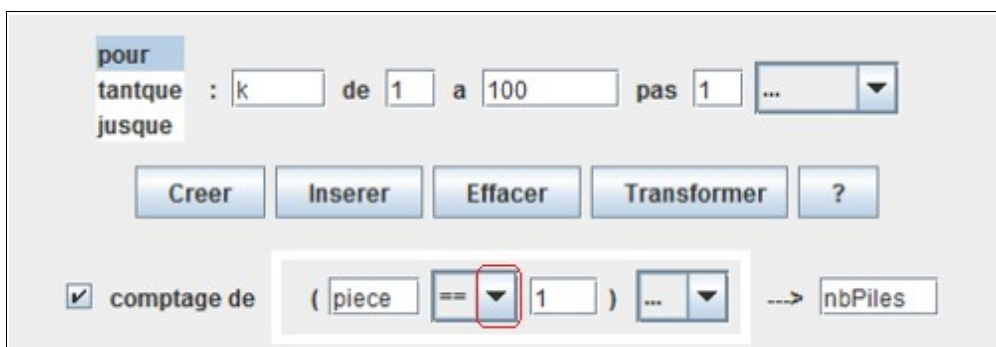
Les programmes obtenus sont exprimés en Larp (langage que j'apprécie particulièrement), mais les indications fournies dans l'article restent valides si l'on choisit un autre langage de développement dans l'onglet Principal.



choix du langage dans l'onglet Principal

Etape 1 : simulation de 100 lancers

Comme indiqué dans l'article principal, l'option « comptage » de l'onglet Boucles permet de compter le nombre de fois où la variable piece (valant 1 ou 2) prend la valeur 1 (pile) :



nombre de piles avec l'option « comptage »

Le résultat obtenu (en cliquant sur **Insérer** ou **Créer**) est à compléter avec une fonction aléatoire renvoyant 1 ou 2 :

```

nbPiles = 0
POUR k=1 JUSQU'A 100 FAIRE
  SI (piece=1) ALORS
    nbPiles = nbPiles+1
  FINSI
FINPOUR

```

piece=ALEATOIRE(1,2)

Etape 2 : répétition de la simulation de 100 lancers

Introduction

Pour répéter la simulation de 100 lancers et compter le nombre de fois où il y aura au moins 60 piles, on peut utiliser l'option « comptage » comme précédemment : il n'est donc pas nécessaire de fournir aux étudiants d'informations supplémentaires sur PluriAlgo.

Néanmoins, nous introduirons dans une seconde approche le bouton **Transformer** de l'onglet Boucles, qui facilite la **répétition** d'instructions sélectionnées dans l'éditeur (les instructions de l'étape 1 ici). Nous parviendrons même directement au résultat final en associant répétition et comptage...

Approche 1 : sans information supplémentaire sur PluriAlgo

Si l'on veut aider les étudiants (en difficulté) à démarrer, on peut leur suggérer de compléter l'onglet Boucles afin de créer une boucle répétant (n fois) la simulation de 100 lancers, tout en calculant (dans la variable nbAuMoins60) le nombre de simulations où le nombre de piles est supérieur à 60 :

The screenshot shows the 'pour' loop configuration in the PluriAlgo editor. The 'pour' button is highlighted in blue. The configuration is as follows:

- pour** (highlighted in blue)
- tantque** : i [] de 1 a n [] pas 1 [] ... [v]
- jusque**
- comptage de** (nbPiles >= 60) ... [v] --> nbAuMoins60

Le résultat obtenu (en cliquant sur le bouton **Insérer**) est le suivant :

```

nbAuMoins60 = 0
POUR i=1 JUSQU'A n FAIRE
  SI (nbPiles>=60) ALORS
    nbAuMoins60 = nbAuMoins60+1
  FINSI
FINPOUR

```

Il reste à y insérer les instructions de l'étape 1 :

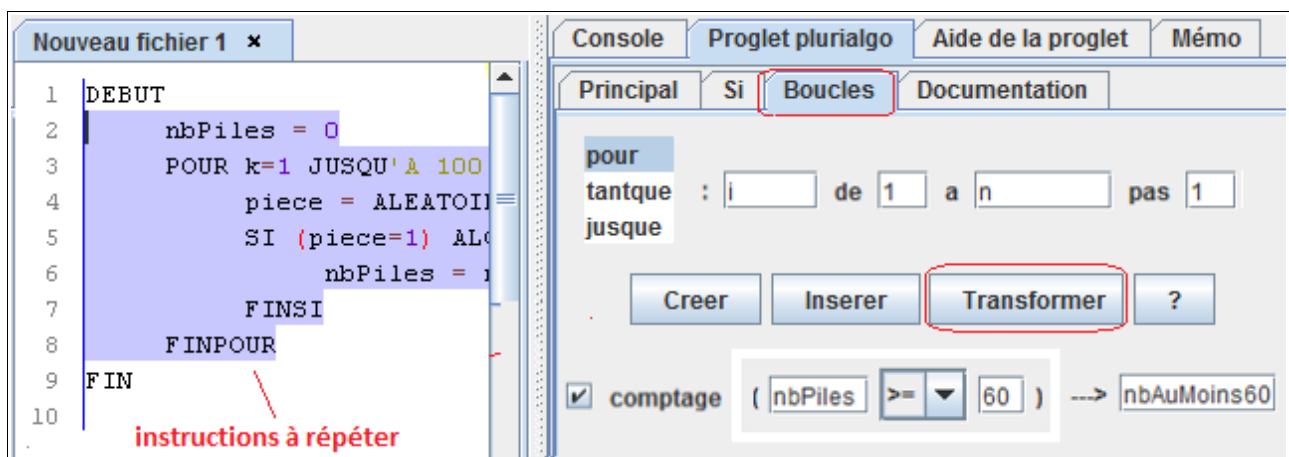
```

nbAuMoins60 = 0
POUR i=1 JUSQU'A n FAIRE
  nbPiles = 0
  POUR k=1 JUSQU'A 100 FAIRE
    piece = ALEATOIRE(1,2)
    SI (piece=1) ALORS
      nbPiles = nbPiles+1
    FINSI
  FINPOUR
  SI (nbPiles>=60) ALORS
    nbAuMoins60 = nbAuMoins60+1
  FINSI
FINPOUR

```

Approche 2 : avec le bouton Transformer

Le programme ci-dessus peut être directement obtenu à partir de l'onglet Boucles, en cliquant sur le bouton **Transformer** :



Ce bouton facilite la **répétition** d'instructions sélectionnées dans l'éditeur (les instructions de l'étape 1 ici) : en effet, elles sont insérées dans la boucle créée par PluriAlgo (Pour i=1 jusqu'à n).

Plus intéressant encore, il est possible d'associer la répétition aux options de l'onglet Boucles (somme, comptage...). Dans le cas présent, cela permet d'obtenir directement le résultat final grâce à l'option « comptage ».