

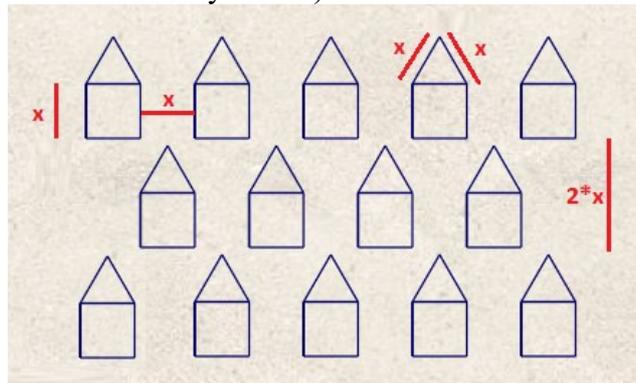
Sous-programmes avec la tortue de Sofus

<http://revue.sesamath.net/spip.php?article932>

patrick.raffinat@univ-pau.fr

A) Introduction

Dans le numéro 53 de MathémaTICE, vous trouverez les énoncés de plusieurs séances de travaux pratiques, dont celui de la semaine 5 (sous-programmes avec la tortue). Dans l'exercice 2 de cette séance fondamentale, mes étudiants devaient aboutir à un programme Blockly dessinant ce lotissement (dont les maisons sont très stylisées !) :



La séance avait en partie été préparée en cours avec des notations algorithmiques :

- définition des procédures « carré » et « triangle », puis d'une procédure « maison » les utilisant.
- utilisation de la procédure « maison » pour afficher une maison, puis pour afficher une rangée de 5 maisons à l'aide d'une boucle.

B) Première phase : procédures « carré », « triangle » et « maison »

La première phase de l'exercice consistait à finaliser avec Blockly le travail de préparation effectué en cours.

Triangles et carrés

La procédure « triangle » était donnée en Blockly lors du premier exercice du TP5, où il fallait faire diverses figures utilisant des triangles.

définition de la procédure

utilisation de la procédure

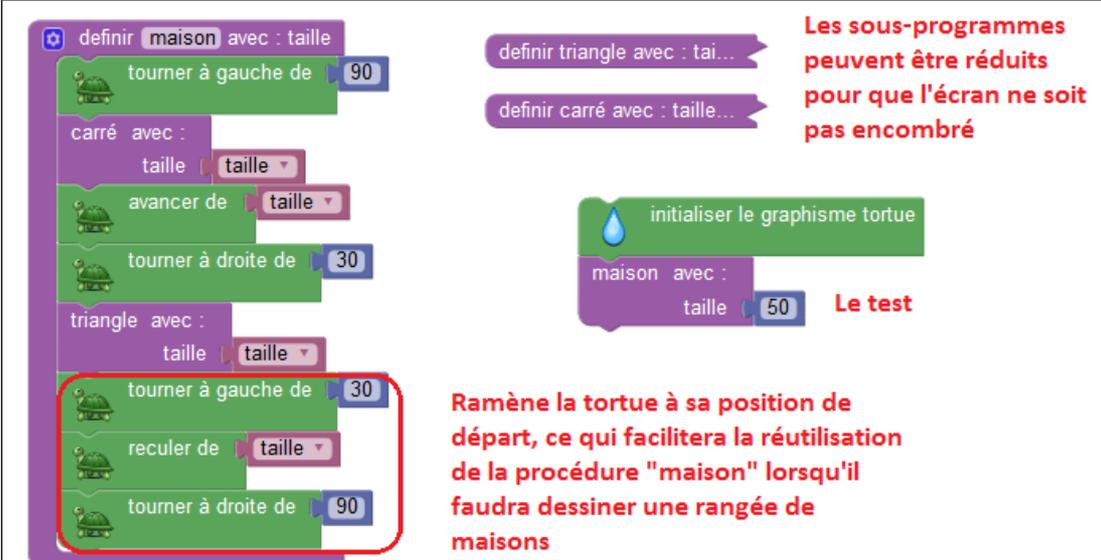
60
80

La procédure « carré » pouvait être définie rapidement grâce à un copier-coller.

Maisons

Dans le cours, j'avais signalé l'importance de choisir des conventions claires facilitant la l'utilisation de la procédure maison :

- maison dessinée en partant du coin inférieur gauche, avec une tortue orientée vers la droite.
- tortue revenant à sa position initiale à la fin de l'exécution, et orientée dans la même direction qu'au départ : cette convention n'est pas fondamentale si on ne dessine qu'une maison, mais le devient quand on en dessine 5.



The image shows a Scratch code editor with several blocks. On the left, a procedure named 'maison' is defined with a parameter 'taille'. The blocks are: 'tourner à gauche de 90', 'carré avec : taille' (containing 'avancer de taille'), 'tourner à droite de 30', 'triangle avec : taille' (containing 'tourner à gauche de 30', 'reculer de taille', 'tourner à droite de 90'). On the right, there are two smaller procedure definitions: 'definir triangle avec : tai...' and 'definir carré avec : taille...'. Below these is a 'Le test' block: 'initialiser le graphisme tortue', 'maison avec : taille' (set to 50). A red box highlights the final three blocks of the 'maison' procedure. Red text annotations explain the purpose of the sub-procedures and the return to the starting position.

Les sous-programmes peuvent être réduits pour que l'écran ne soit pas encombré

Ramène la tortue à sa position de départ, ce qui facilitera la réutilisation de la procédure "maison" lorsqu'il faudra dessiner une rangée de maisons

Avant de passer à 5 maisons, j'ai demandé aux étudiants de faire un test pour voir si la maison était correctement dessinée et si la tortue reprenait sa position initiale, ce qui a donné lieu à de nombreux ajustements.

C) Deuxième phase : une rangée de maisons

Une rangée de 5 maisons

Comme cela avait été préparé en cours avec des notations algorithmiques, cela n'a pas posé de problème particulier, si ce n'est qu'il a fallu réduire la taille des maisons (50) pour que certaines maisons ne sortent pas du cadre, une autre solution étant de déplacer vers la gauche la tortue avant de faire le dessin.



The image shows a Scratch code editor with a sequence of blocks. It starts with 'initialiser le graphisme tortue'. Then, a red box highlights three blocks: 'lever le stylo', 'reculer de 300', and 'baisser le stylo'. To the right of this box, red text says 'tortue déplacée vers la gauche'. Below this is a 'répéter 5 fois' loop containing a 'faire' block with 'maison avec : taille' (set to 50), 'lever le stylo', 'avancer de 50 x 2', and 'baisser le stylo'.

tortue déplacée vers la gauche

Procédure « rangéeDeMaisons »

The image shows two Scratch code snippets. The left snippet is a procedure definition for 'rangéeDeMaisons' with parameters 'taille' and 'nombre_maison'. It contains a 'répéter' block with 'nombre_maison' iterations, and a 'faire' block with 'lever le stylo', 'avancer de' (taille * 2), and 'baisser le stylo'. A red box highlights the 'nombre_maison' parameter and the 'taille' parameter in the 'faire' block, with a red arrow pointing to a note: "Des étudiants ont laissé le 5 et le 50 de la question précédente". Below the 'faire' block, another red box highlights 'lever le stylo', 'reculer de' (taille * nombre_maison * 2), and 'baisser le stylo', with a note: "on ramène la tortue à sa position initiale cela facilitera l'affichage de plusieurs rangées". The right snippet shows the procedure being called with 'rangéeDeMaisons avec : taille 50 nombre_maison 5', with red boxes around the values 50 and 5, and a note: "Il faut veiller à ce que les étudiants fassent aussi des tests avec d'autres valeurs".

Prudemment, j'avais indiqué dans l'énoncé que la procédure « rangéeDeMaisons » devait être paramétrée par la taille d'une maison et le nombre de maisons. Mais certains étudiants n'ont pas tenu compte des paramètres dans le corps du sous-programme : donc, « rangéeDeMaisons(50,5) » affichait 5 maisons, tout comme « rangéeDeMaisons(50,4) » ! Ce n'était pas une étourderie, puisque ces étudiants faisaient alors appel à moi, ne trouvant pas la cause de leur erreur...

D) Troisième phase : le lotissement

La procédure « rangéeDeMaisons » devait être appliquée 3 fois, avec un déplacement adéquat de la tortue entre chaque rangée :

The image shows two Scratch code snippets. The left snippet is a procedure definition for 'rangéeDeMaisons' with parameters 'x' and 'nombre_maison'. It contains a 'fixer x à' block with '30', and a 'rangéeDeMaisons avec : taille x nombre_maison 5' block. The 'rangéeDeMaisons' block is followed by 'lever le stylo', 'avancer de' (x), 'tourner à droite de' (90), 'avancer de' (x * 2), 'tourner à gauche de' (90), and 'baisser le stylo'. The 'rangéeDeMaisons' block is followed by 'rangéeDeMaisons avec : taille x nombre_maison 4'. The right snippet shows the procedure being called with 'rangéeDeMaisons avec : taille x nombre_maison 5', with red boxes around the 'x' parameter and the value 5, and a note: "Il faut veiller à ce que les étudiants fassent aussi des tests avec d'autres valeurs".

Le fichier Blockly final est disponible dans l'article principal : vous pouvez le télécharger (bouton Ouvrir de Mathém@ALGO) et le tester (bouton Exécuter), après l'avoir dézippé.

E) Petit extra pour amateurs de Python

Le programme Blockly peut être exporté en Python (bouton « Exporter Python »), pour être ensuite testé dans un environnement externe :

| | |
|---|---|
| <pre># code Python cree par Blockly from turtle import * taille = None nombre_maison = None x = None def triangle(taille): for count in range(3): forward(taille) right(120) def carre(taille): for count2 in range(4): forward(taille) right(90) def maison(taille): left(90) carre(taille) forward(taille) right(30) triangle(taille) left(30) backward(taille) right(90)</pre> | <pre>def rangeeDeMaisons(taille, nombre_maison): for count3 in range(int(nombre_maison)): maison(taille) penup() forward((taille * 2)) pendown() penup() backward((taille * (nombre_maison * 2))) pendown() reset() x = 30 rangeeDeMaisons(x, 5) penup() forward(x) right(90) forward((x * 2)) left(90) pendown() rangeeDeMaisons(x, 4) penup() backward(x) right(90) forward((x * 2)) left(90) pendown() rangeeDeMaisons(x, 5)</pre> |
|---|---|

Le code Python peut même être testé avec Mathém@ALGO !

The screenshot shows the 'Execution' tab of the Mathém@ALGO interface. It contains the following text:

Vous pouvez exécuter des programmes Javascript en cliquant sur le bouton **Executer**.

Autres options : Javascript [tableur](#), langage [Python](#), langage [Processing](#), Javascript [multi-tortues](#), script [DGPad](#).

Pensez à cliquer sur le lien Python avant de cliquer sur le bouton Executer

At the bottom, there is a button labeled 'Executer' which is highlighted with a red box.