

Programmation : variables, boucles et blocs personnalisés

I - Lire et décrire un assemblage de blocs

Définition 1

Un **script** est une suite d'instructions exécutées dans l'ordre, de haut en bas. Un **programme** est l'ensemble des scripts d'un projet.

Vocabulaire

Un bloc porte une couleur selon sa catégorie : événements, mouvement, contrôle, variables, capteurs. La forme dit comment il s'assemble — un bloc en C contient d'autres blocs.

Méthode 1 — Prévoir le résultat d'un script

1. Lister les variables et noter leur valeur de départ.
2. Exécuter les instructions une à une, en tenant à jour un tableau des valeurs.
3. Ne jamais sauter une instruction : l'ordinateur n'en saute aucune.

Exemple — Une trace d'exécution

Pour le script « mettre a à 5 ; mettre b à 3 ; mettre a à $a + b$; mettre b à $a - b$ » :

Instruction	a	b
mettre a à 5	5	—
mettre b à 3	5	3
mettre a à $a + b$	8	3
mettre b à $a - b$	8	5

II - Variables informatiques

Définition 2

Une **variable** informatique est une case mémoire, désignée par un nom, qui contient une valeur. Cette valeur peut changer au cours de l'exécution.

L'instruction « mettre a à $a + 1$ » n'est pas une équation : elle se lit « remplacer le contenu de a par sa valeur augmentée de 1 ». Le signe d'affectation va de la droite vers la gauche.

Méthode 2 — Utiliser une variable

1. La créer et lui donner un nom qui dit son rôle : score, côté, distance.

2. L'initialiser avant la première utilisation.
3. La modifier avec « mettre ... à ... » ou « ajouter ... à ... ».

Remarque

Oublier l'initialisation est l'erreur la plus fréquente : la variable garde alors la valeur de l'exécution précédente, et le programme ne donne pas deux fois le même résultat.

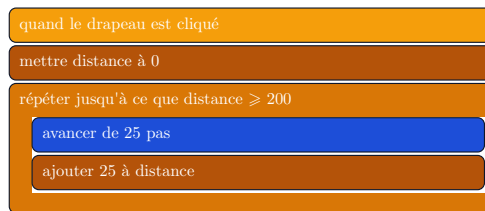
III - Les boucles

Définition 3

Une **boucle bornée** « répéter n fois » exécute un groupe d'instructions un nombre de fois connu à l'avance.

Définition 4

Une **boucle non bornée** « répéter jusqu'à ce que ... » exécute un groupe d'instructions tant qu'une condition n'est pas réalisée. Le nombre de tours n'est pas connu à l'avance.



une boucle non bornée : le nombre de tours dépend de la condition

Fig. 1. – Une boucle non bornée : elle s'arrête quand la condition devient vraie.

Si la condition d'arrêt n'est jamais réalisée, le programme tourne indéfiniment. Toute boucle non bornée demande de vérifier que son corps fait progresser vers l'arrêt.

Exemple — Un polygone régulier

Pour tracer un polygone régulier à n côtés, on répète n fois : avancer de la longueur du côté, puis tourner de $\frac{360}{n}$ degrés. La somme des rotations vaut alors un tour complet.

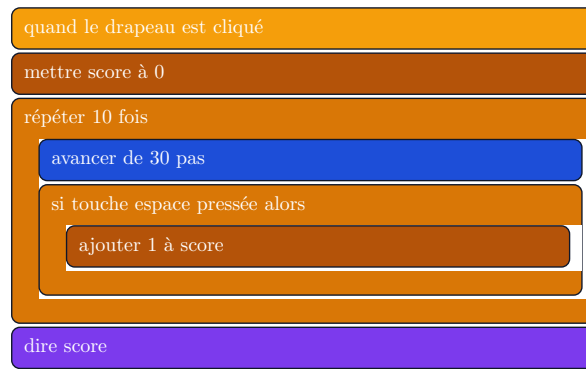
Méthode 3 — Construire une figure par répétition d'un motif

1. Isoler le motif élémentaire, celui qui se répète.
2. L'écrire une fois, en s'assurant qu'il ramène à la position de départ.
3. L'enfermer dans une boucle, en ajoutant le déplacement ou la rotation qui sépare deux motifs.

IV - Les instructions conditionnelles

Définition 5

Une **instruction conditionnelle** « si ... alors » n'exécute son contenu que lorsque la condition est vraie. La forme « si ... alors ... sinon » propose deux issues exclusives.



une boucle bornée, contenant une instruction conditionnelle

Fig. 2. – Une boucle bornée contenant un test : à chaque tour, le score n’augmente que si la touche est pressée.

Exemples — Des conditions

- si score ≥ 10 alors dire « gagné »
- si la couleur rouge est touchée alors mettre vies à vies - 1
- si $x < 0$ alors dire « négatif » sinon dire « positif ou nul »

Remarque

Une condition est une affirmation qui n’a que deux valeurs possibles : vraie ou fausse. C’est ce qui distingue le test = de l’affectation.

V - Décomposer un problème : le bloc personnalisé

Un programme long devient illisible. La solution est la même qu’en mathématiques : découper le problème en sous-problèmes, et donner un nom à chacun.

Définition 6

Un **bloc personnalisé** est une suite d’instructions à laquelle on donne un nom. Une fois défini, il s’utilise comme n’importe quel autre bloc, autant de fois que nécessaire.



à gauche le bloc défini une fois, à droite le script qui l’appelle

Fig. 3. – Le bloc carré est défini une seule fois, avec un paramètre ; le script principal l’appelle six fois pour dessiner une rosace.

Définition 7

Un **paramètre** est une variable propre au bloc, dont la valeur est fournie au moment de l’appel. C’est lui qui rend le bloc réutilisable.

Méthode 4 — Créer un bloc personnalisé

1. Repérer un groupe d’instructions qui revient, ou qui a un sens à lui seul.

2. Lui donner un nom qui dit ce qu'il fait, à l'infinitif ou par un nom d'objet.
3. Ajouter un paramètre pour ce qui change d'un appel à l'autre.
4. Remplacer chaque occurrence par un appel au bloc.

Remarque

Un bloc personnalisé rend le programme plus court, plus lisible, et surtout plus facile à corriger : la modification se fait à un seul endroit.

VI - Scripts en parallèle et évènements

Définition 8

Un **évènement** déclenche un script : le clic sur le drapeau, l'appui sur une touche, la réception d'un message, un contact entre deux objets.

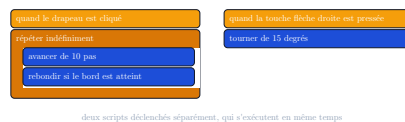


Fig. 4. – Deux scripts indépendants, déclenchés par deux évènements différents et actifs en même temps.

Définition 9

Deux scripts s'exécutent **en parallèle** lorsqu'ils avancent tous les deux, chacun à son rythme, sans que l'un attende l'autre.

Exemple — Un jeu simple

- Script 1 : quand le drapeau est cliqué, répéter indéfiniment, avancer et rebondir sur les bords.
- Script 2 : quand la flèche droite est pressée, tourner de 15 degrés.
- Script 3 : quand le lutin touche le bord rouge, jouer un son et arrêter tout.

Les trois cohabitent : le premier fait vivre le jeu, les deux autres réagissent aux actions du joueur.

Méthode 5 — Faire communiquer deux scripts

- Par un **message** : un script envoie « partie perdue », un autre réagit à sa réception.
- Par une **variable partagée** : un script met vies à 0, un autre teste si `vies = 0` alors.

Remarque

Le parallélisme rend les erreurs plus difficiles à trouver : deux scripts qui modifient la même variable peuvent se gêner. Une variable ne devrait être modifiée que par un seul script.