

Petite introduction à la programmation événementielle

IREM d'Aquitaine – Groupe Algorithmique

IREM d'Aquitaine – mai 2017

Les « paradigmes » de programmation

Un paradigme de programmation est un « style de programmation » qui précise de quelle façon le développeur (programmeur) doit mettre en œuvre sa solution (son programme).

Chaque paradigme de programmation peut être supporté par différents langages de programmation :

- Programmation séquentielle : Python, C, C++, Java, Pascal, Fortran...
- Programmation événementielle : Scratch, Visual Basic, Simula...
- Programmation orientée objet : Python, C++, Java, Smalltalk...
- Programmation concurrente : C++, Java, Go...
- Programmation fonctionnelle : Lisp, Caml, Ruby...
- Programmation logique : Prolog...
- etc.

IREM d'Aquitaine – mai 2017

2

La programmation séquentielle

Le paradigme de programmation séquentielle est le paradigme de programmation le plus « traditionnel » (parfois appelé impératif, ou procédural) :

Le programme s'exécute étape par étape, dans un ordre contrôlé par le programmeur :

- première instruction = début du programme
- enchaînement défini par les structures de contrôles utilisées :
 - enchaînement séquentiel (par défaut),
 - structures alternatives (du type si-alors-sinon)
 - structures répétitives (boucles pour et tant-que)
 - appel de sous-programmes (fonctions)
 - ...

IREM d'Aquitaine – mai 2017

3

La programmation séquentielle (un exemple)

Un programme Python affichant la liste des nombres premiers inférieurs à 100 :

```
# ce programme affiche la liste des nombres premiers inférieurs à 100
import math
for n in range(1,100):
    # initialisations
    racineDeN = int(math.sqrt(n))
    diviseur = 2
    # tant qu'on n'a pas trouvé de diviseur, on avance...
    while ((n % diviseur != 0) and (diviseur <= racineDeN)):
        diviseur = diviseur + 1
    # si diviseur est allé au-delà de racineDeN, N est premier
    if (diviseur > racineDeN):
        print(n)
```

IREM d'Aquitaine – mai 2017

4

La programmation événementielle

Le programme est constitué de « portions de code », qui seront exécutées en fonction de la survenue d'événements particuliers.

Exemples d'événements :

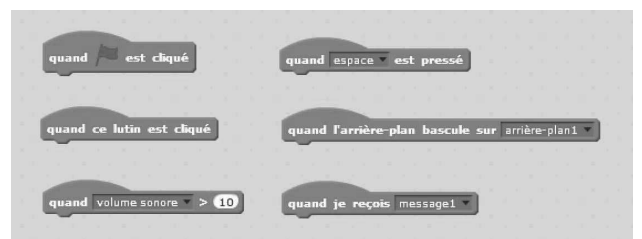
- lancement du programme,
- appui sur une touche clavier,
- action sur la souris : clic gauche, clic droit, double-clic, déplacement...
- message reçu (par une autre portion de code),
- ...

Le programmeur « se contente » d'écrire les portions de code associées aux événements. La gestion des événements (appel des portions associées) est entièrement automatisée...

IREM d'Aquitaine – mai 2017

5

Quelques événements Scratch



La portion de code associée est « emboîtée » sous l'événement correspondant...

IREM d'Aquitaine – mai 2017

6

Synchronisation par échange de messages

Outre les réponses aux événements déclenchés par l'utilisateur du programme (lancement, touches clavier, souris...), les lutins Scratch peuvent « réagir » (exécuter une portion de code) à la réception de messages... envoyés par d'autres lutins !

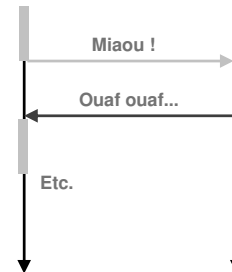


Cela va notamment permettre aux lutins de « synchroniser leurs actions »...

Illustration : petit dialogue entre chien et chat



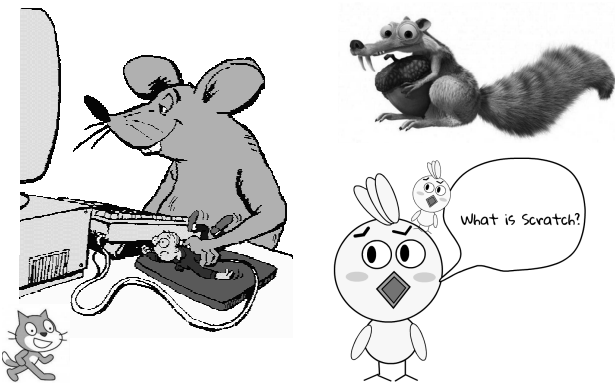
Chien et chat




Deux remarques :

- un lutin peut « continuer à travailler » après l'envoi d'un message...
- plusieurs lutins peuvent « se mettre à travailler » lors de la réception d'un même message, simultanément...

Merci de votre attention...



FICHE ÉLÈVE

Scratch offre la possibilité de dessiner à l'écran en affichant la trace du lutin lors de ses déplacements. Il faut pour cela utiliser le menu  Stylo .

1.
 - a. Ecrire un script permettant d'abaisser le stylo lorsqu'on appuie sur la touche « E ».
 - b. Ecrire un script permettant de lever le stylo lorsqu'on appuie sur la touche « L ».
 - c. Ecrire un script permettant de nettoyer la scène lorsqu'on appuie sur la touche « N ».
 - d. Ecrire un script permettant d'augmenter de 1 la taille du stylo lorsqu'on appuie sur la touche « P ».
 - e. Ecrire un script permettant de diminuer de 1 la taille du stylo lorsqu'on appuie sur la touche « M ».

2. Écrire un script permettant au lutin de tracer le chiffre 2 au format « réveil matin » lorsqu'on appuie sur la touche « 2 ».

3. Écrire d'autres scripts permettant d'écrire les autres chiffres au même format.

FICHE PROF

Le but de cette activité est de découvrir les fonctionnalités de dessin du logiciel.

1. On demande aux élèves d'écrire des scripts permettant de contrôler le fait d'abaisser ou de relever le stylo, d'effacer le dessin ou de modifier la taille du stylo. Ces scripts seront déclenchés chacun par une touche dédiée du clavier :

Touche E : Écrire

Touche L : Lever le stylo

Touche N : Nettoyer la scène

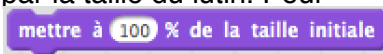
Touche P : augmenter de 1 la taille du stylo

Touche M : diminuer de 1 la taille du stylo

2. On demande maintenant aux élèves de tracer le chiffre 2 au format « réveil matin ». Ils vont pour cela mêler leurs connaissances sur les déplacements, acquises avec la fiche *Déplacements* et les scripts de la question 1. .

Un exemple de solution :



Remarque : Une partie de la figure se retrouve parfois masquée par la taille du lutin. Pour améliorer cela, il suffit, dans le menu *Apparence*, d'utiliser le bloc  avec une valeur de 1%.

3. Enfin, on demande aux élèves d'écrire les scripts correspondants aux autres chiffres.

Remarque : Certains élèves vont être bloqués ou gênés par le fait de devoir repasser sur le même trait. Ces passages multiples sont sans incidence sur le résultat final et, bien qu'il soit possible de réaliser les figures sans repasser sur le même trait, cela s'avère plus complexe avec plus d'actions de relever et abaisser le stylo.

FICHE ÉLÈVE

Le but de cette activité est de déplacer le personnage à l'écran à l'aide de touches du clavier de l'ordinateur.

1. Choisis un nouveau lutin en cliquant sur l'icône : . Il s'agit de la souris nommée Mouse1.




2. Écris un script permettant de faire avancer la souris en appuyant sur la touche « Flèche droite ». Teste-le en appuyant plusieurs fois sur la touche « Flèche droite ».
3. Écris un autre script permettant de faire avancer la souris dans la direction opposée en appuyant sur la touche « Flèche gauche ». Teste-le en appuyant plusieurs fois sur la touche « Flèche gauche ».
4. Complète ton programme avec deux autres scripts afin que la souris se déplace dans les quatre directions à l'aide des quatre touches « Flèches ». Teste les scripts en appuyant plusieurs fois sur les touches « Flèche haut » et « Flèche bas ».

APPELLE LE PROFESSEUR.

5. Colle à présent chacun de tes scripts dans la page de scripts du chat. Pour cela, glisse le script jusque sur le lutin choisi (celui dans la liste en bas de la scène) et relâche le bouton de la souris. Que remarques-tu quand tu déplaces le lutin à l'aide de la touche « flèche gauche » ?
6. Modifie les scripts afin que le lutin se déplace en style « à 360° » pour les déplacements vers le haut, le bas et la droite et en style « position à gauche ou à droite » pour le déplacement vers la gauche.

FICHE PROF

Cette activité est une initiation aux déplacements du lutin à l'aide des touches flèches du clavier.

On va pour cela utiliser le lutin « Mouse1 ». Les élèves doivent aller chercher ce nouveau lutin en cliquant sur l'icône .




Le but de l'activité étant de faire découvrir aux élèves les mouvements du lutin, ils vont devoir réaliser que deux quantités entrent en jeu : la direction du lutin et sa distance de parcours.

On commence par proposer aux élèves d'écrire un script permettant de faire « avancer » le lutin en appuyant sur la touche « flèche droite ».

Production probable des élèves :



On leur demande ensuite d'écrire un autre script permettant de faire avancer le lutin dans la direction opposée en appuyant sur la touche « flèche gauche ». Les réponses sont souvent les suivantes :

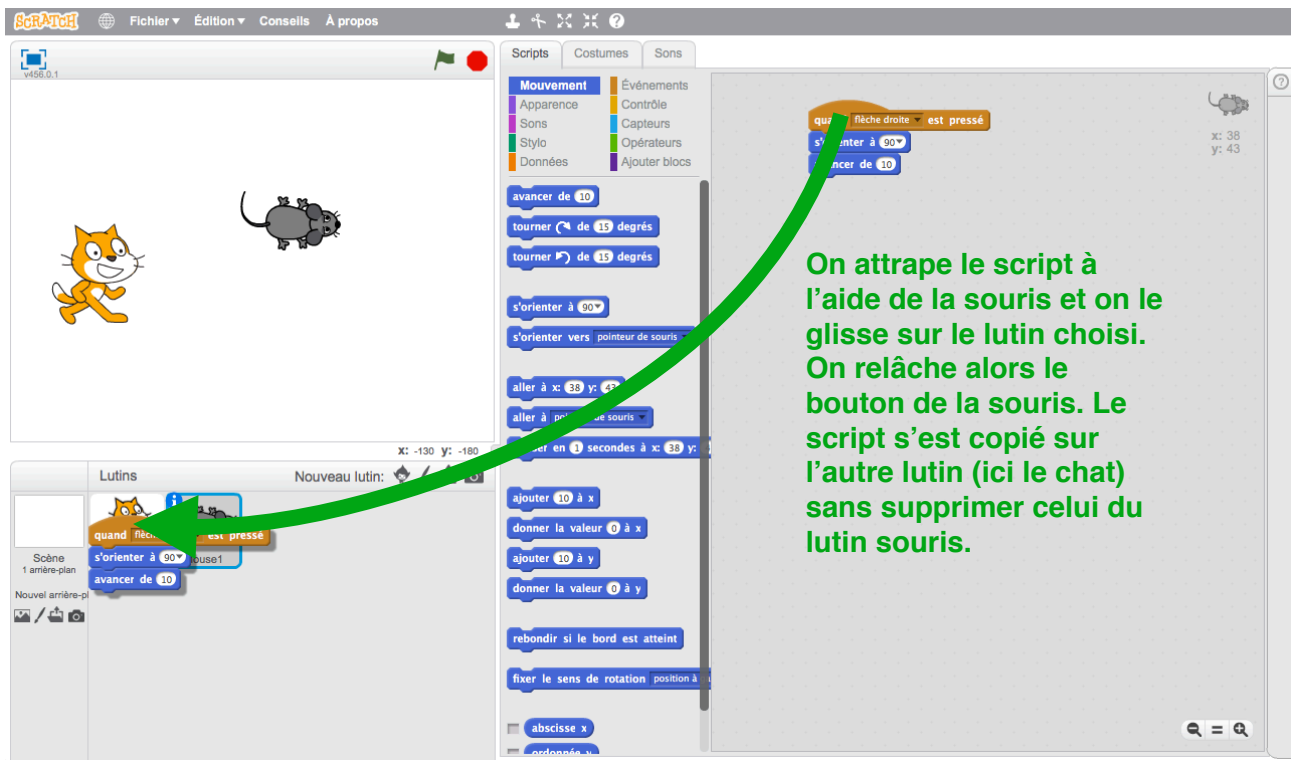
	Les élèves pensent que la touche « flèche gauche » suffit à orienter la souris.	En testant, ils voient d'eux même que cette solution n'est pas bonne.
	La souris se déplace bien vers la gauche mais elle recule au lieu d'avancer.	On peut dire aux élèves que l'on souhaite obtenir un déplacement « naturel » de la souris (c-à-d où la souris avance quand elle se déplace).
	Si l'on n'appuie qu'une fois sur la touche « flèche gauche », la souris se retourne bien et avance vers la gauche. Si l'on appuie une seconde fois, elle se retourne vers la droite puis avance (vers la droite...).	Il est important de demander aux élèves de tester le bon fonctionnement du script en exécutant plusieurs fois de suite celui-ci.

Maintenant que les élèves ont compris que l'on doit orienter le lutin avant de le faire avancer, on peut leur demander de créer deux scripts pour déplacer le lutin vers le haut et vers le bas.

Remarque : une fois terminé, les élèves doivent appeler le professeur. Ce moment de vérification est important car certains élèves positionnent le bloc « s'orienter à » dans les trois directions haut, bas et gauche mais omettent de le rajouter également dans le script déclenché par la touche droite.

FICHE PROF

Les déplacements de la souris étant terminés, on demande aux élèves de copier-coller les scripts dans le lutin chat.



Ils observent alors que lorsque l'on appuie sur la touche « flèche gauche », le lutin se déplace avec la tête en bas.

Il est temps de faire un interlude pour présenter le menu information du lutin (i). Il contient notamment la valeur de la direction du lutin ainsi que le **style de rotation**.

- **style de rotation:** ↔ ● l'image du lutin « suit » la direction :

direction: -131° affichage sur la scène →

- **style de rotation:** ↔ ● l'image du lutin s'oriente à gauche si l'angle est dans l'intervalle] -180 ; 0 [modulo 360. Il s'oriente à droite si l'angle est dans l'intervalle [0 ; 180] modulo 360 :

direction: -131° affichage sur la scène →

- **style de rotation:** ↔ ● l'image du lutin reste fixe quelle que soit la direction de celui-ci.

direction: -131° affichage sur la scène →

FICHE PROF

Info pratique : on peut modifier le style de rotation dans les scripts. Le bloc dévolu à cette fonction se trouve dans le menu *Mouvement* :



fixer le sens de rotation position à gauche ou à droite ▼

Il y a une erreur de traduction dans la langue française. Ce bloc devrait s'intituler « fixer le style de rotation ».


Il est également important de faire remarquer aux élèves la différence entre direction absolue (S'orienter à) et direction relative (Tourner).

Exemple :

Si le chat a une direction de 13° et que l'on exécute le bloc « tourner à droite de 90° », le chat aura une direction de 103° .

Si le chat a une direction de 13° et que l'on exécute le bloc « s'orienter à 90° », le chat aura une direction de 90° .

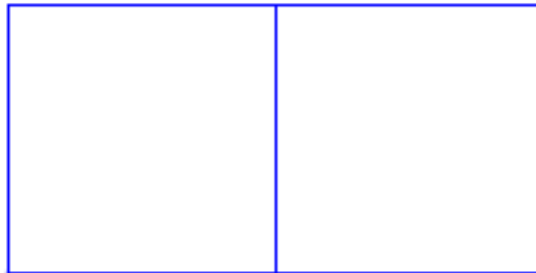
FICHE ÉLÈVE

Le but de cette activité est de réinvestir ses connaissances sur le menu  et de découvrir de nouvelles possibilités d'écritures de scripts.

1. Écris un script permettant de dessiner un carré de côté 100 quand on appuie sur la touche « c ».

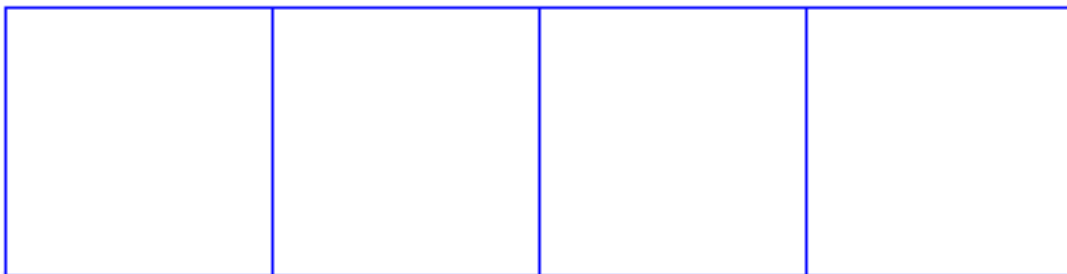
APPELLE LE PROFESSEUR

2. Écris un script permettant de dessiner les deux carrés de côté 100 ci-dessous quand on appuie sur la touche « espace » :



APPELLE LE PROFESSEUR

3. Complète le script précédent afin de dessiner les quatre carrés de côté 100 ci-dessous :



APPELLE LE PROFESSEUR


4. Modifie le script précédent afin de dessiner les huit carrés de côté 50 ci-dessous :



APPELLE LE PROFESSEUR

FICHE PROF

Cette activité est une introduction à l'utilisation des boucles finies.

Le but de l'activité étant d'introduire l'utilisation du bloc , les élèves devront réaliser que des actions identiques sont répétées plusieurs fois.

1. Pour commencer, on propose aux élèves de tracer un carré de côté 100 quand la touche « c » du clavier est pressée.

Pour la plupart des élèves, la fiche 2 aura suffi à lever les difficultés.

Toutefois, pour certains élèves, le lutin tourne en avançant.

Une production possible :



affichage sur la scène →



On peut leur proposer d'écrire le script pas à pas afin de visualiser qu'il n'y a pas de dessin

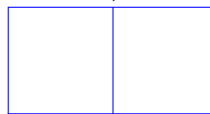
supplémentaire lorsque l'on rajoute le bloc  :



affichage sur la scène →



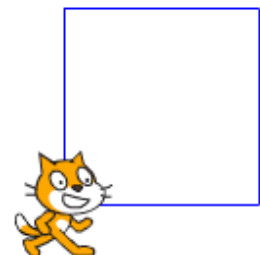
2. On propose maintenant aux élèves de dessiner, à l'aide d'un nouveau script, deux carrés de côtés 100 :



Production possible des élèves :



affichage sur la scène →



FICHE PROF

Ils dupliquent le script du carré une fois **sans faire avancer** le chat entre les deux carrés.
Pour éviter cela, il suffit de leur demander où se trouve le lutin après avoir dessiné le premier carré.

L'insertion d'un bloc supplémentaire  leur paraît alors naturelle.

3. Maintenant que les élèves ont compris que l'on doit faire avancer le lutin avant le tracé d'un nouveau carré, on peut leur demander de compléter leur script afin de dessiner quatre carrés de côté 100 comme ci-dessous :

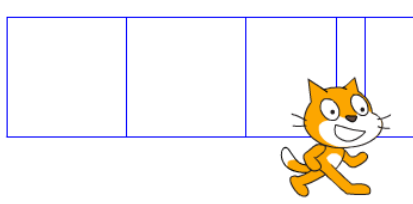


Production possible des élèves :




FICHE PROF

Selon la position du lutin sur l'écran, il est possible que la frise ne se trace pas correctement. Un rendu possible :



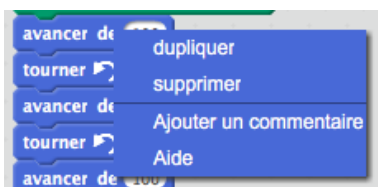
Cela est dû au fait que le lutin « bute » contre le bord de la scène et continue d'avancer (comme un robot contre un mur). Au lieu d'avoir avancé de 100, la longueur du côté s'en trouve donc tronquée.

Pour éviter ce problème de tracé, on peut proposer aux élèves de replacer à la souris le lutin suffisamment vers la gauche de la scène pour que la frise « rentre » dans la scène. On peut

également utiliser le bloc  qui permet de positionner le lutin par rapport à des coordonnées sur la scène en début de script.

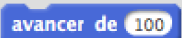
Les élèves trouvent le script long à réaliser. Certains trouvent la parade en utilisant la commande « dupliquer » avec le clic droit de la souris.

Info pratique : le logiciel Scratch offre une fonctionnalité de duplication de blocs. Lorsque l'on effectue un clic droit sur un bloc et que l'on clique sur « Dupliquer », le logiciel duplique celui-ci ainsi que tous les autres blocs qui lui sont attachés dessous.



4. On propose de modifier ce script afin de dessiner huit carrés de côté 50 comme ci-dessous :



Même pour les élèves qui ont utilisé la fonction « dupliquer », cette phase est très fastidieuse. Il s'agit ici de modifier chaque bloc  en changeant la valeur 100 par 50.

Il faut ensuite dupliquer encore une fois tout le script afin d'obtenir 8 carrés et non 4 comme précédemment.

À ce moment de l'activité, on peut leur faire formuler qu'ils vont devoir répéter un carré huit fois. Il est important que le verbe « répéter » émerge.

Dès lors, on peut leur suggérer de chercher un bloc qui les aiderait à répéter ce carré.

FICHE PROF

Production possible des élèves :



affichage sur la scène →



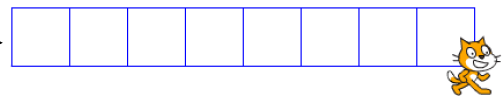
Pour aller plus loin :

On peut demander aux élèves les plus à l'aise d'optimiser ce script afin d'éviter toute répétition.

Production possible des élèves :



affichage sur la scène →



Remarque : Lors de l'exécution de ce dernier script, la double répétition engendre une plus grande lenteur dans le tracé, ce qui permet à l'élève de mieux le visualiser.