

# Notions informatiques et repères de progressivité pour le cycle 4

Groupe IREM "Algorithmes et programmation au cycle 4" de Nantes

Christophe Declercq

---

Collège informatique de Nantes – Ecole d'été 27 juin 2018

Groupe IREM Algorithmes et programmation au cycle 4 – Christophe Declercq

## Thème E - Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

Attendus de fin de cycle

Écrire, mettre au point et exécuter un programme simple

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
<p>Décomposer un problème en sous-problèmes afin de structurer un programme ; reconnaître des schémas.</p> <p>Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné.</p> <p>Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.</p> <p>Programmer des scripts se déroulant en parallèle.</p> <p>» Notions d'algorithme et de programme.</p> <p>» Notion de variable informatique.</p> <p>» Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles.</p>	<p>Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe.</p> <p>Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle.</p> <p>Initiation au chiffrement (Morse, chiffre de César, code ASCII...).</p> <p>Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis...</p> <p>Calculs simples de calendrier.</p> <p>Calculs de répertoire (recherche, recherche inversée...).</p> <p>Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.</p>

Repères de progressivité

En 5<sup>e</sup>, les élèves s'initient à la programmation événementielle. Progressivement, ils développent de nouvelles compétences, en programmant des actions en parallèle, en utilisant la notion de variable informatique, en découvrant les boucles et les instructions conditionnelles qui complètent les structures de contrôle liées aux événements.

# Contexte

## ■ Thème E – Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

Exemples d'activités possibles : jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe, jeu du cadavre exquis.

### Attendus de fin de cycle

- Écrire, mettre au point et exécuter un programme simple.

### Écrire, mettre au point, exécuter un programme

#### Connaissances

- Notions d'algorithme et de programme
- Notion de variable informatique

#### Compétences associées

- Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné

Collège informatique de Nantes – Ecole d'été 27 juin 2018

Groupe IREM Algorithmes et programmation au cycle 4 – Christophe Declercq

# Objectifs

- Distinguer les concepts à enseigner des concepts savants
- Isoler différents "niveaux de complexité", pour en déduire des repères de progressivité
- Traiter l'ensemble des notions communes aux programmes de cycle 4 mathématiques et technologie

# Notions (1)

- **Algorithme** : description d'un procédé pour calculer quelque chose, pour résoudre un problème. Un algorithme est écrit pour un interlocuteur humain pour lui expliquer comment on va procéder.
- **Programme** : traduction d'un algorithme dans un langage compréhensible par une machine

# Notions (2)

- **Machine** : dispositif technique - le plus souvent électronique - capable d'exécuter les instructions d'un programme enregistré en mémoire. Un ordinateur dispose au moins d'un dispositif de calcul, et d'une mémoire dans laquelle sont enregistrées données et programmes.
- **Langage** : permet de mémoriser ou communiquer des informations en utilisant un ensemble de signes. Un langage de programmation est défini par son lexique (l'ensemble des mots clés et symboles du langage), sa grammaire (les règles de construction d'un programme) et son sens (qui détermine l'effet de l'exécution des programmes).

# Notions (3)

- **Information** : quelque chose que l'on connaît - une connaissance - sous une forme qui permettra de s'en souvenir - la mémoriser - ou de la communiquer à quelqu'un d'autre.
- Les informations pour être traitées par un ordinateur doivent être codées.
- **Informatique** : science du traitement de l'information.

# Structuration des algorithmes et des programmes

- Algorithmes et programmes sont écrits en utilisant des instructions élémentaires et en les **composant / structurant** ensemble pour obtenir des algorithmes ou programmes plus complexes.
- Les **instructions élémentaires** dépendent de l'environnement ou du micro-monde utilisé.

Dans un monde "quadrillé", ce peut être un jeu d'instructions de déplacements absolus : Nord, Sud, Est, Ouest ou relatifs : Avance, Recule, Tourne Droite, Tourne Gauche

Dans un environnement de programmation de calcul, ce peut être : Ajoute, Multiplie, Stocke, Rappel mémoire

# Séquence

- La notion la plus élémentaire et la plus fondamentale.  
Programmer c'est "anticiper" c'est à dire planifier la "séquence"  
- dans l'ordre - des actions à mettre en oeuvre pour résoudre  
une situation
- Situation type : programmer le déplacement d'un personnage  
ou d'un robot par enchaînement - séquentiel - de déplacements  
élémentaires.

# Séquence – repères de progressivité

- 1) Situation où le déroulement de l'exécution montre tous les états intermédiaires
- 2) Situation définie par un état initial et un état final, dont les étapes intermédiaires ne sont pas visibles à l'exécution mais sont spécifiées par la donnée du problème.
- 3) Situation où l'élève doit inventer des étapes intermédiaires pour aboutir à la résolution du problème

# Répétitions

- La complexité d'une situation nécessitant l'usage d'une boucle dépend de la difficulté intrinsèque de cette boucle, mais aussi de la nature de ce qui sera à construire à l'intérieur
- Compétences :
  - savoir évaluer l'exécution d'un programme comportant une ou des boucles consiste à savoir montrer, pour une situation particulière, la séquence d'actions correspondant au déroulement de la boucle.
  - savoir résoudre un problème nécessitant de programmer une ou des boucles consiste à identifier un traitement élémentaire pouvant être répété un certain nombre de fois, donné à l'avance ou déterminé à l'exécution.

# Répétition – repères de progressivité

- 1) **Nombre d'itérations constant**. Forme : répéter 4 fois  
Situation type : écrire de manière plus concise, un programme contenant une séquence d'instructions identiques.
- 2) Nombre d'itérations borné. Forme : répéter n fois
- 3) **Avec indice de boucle**. Forme : pour i variant de 1 à n
- 4) Nombre d'itérations non borné. Forme : répéter indéfiniment
- 5) Boucle avec condition continuation/arrêt.  
Forme : répéter tant que, répéter jusqu'à

# Répétition – repères de progressivité (suite)

- Pour chacune des boucles, on pourra aussi distinguer les cas plus simples avec une seule instruction élémentaire à répéter, des cas plus complexes avec une séquence d'instructions ou une instruction complexe pouvant elle-même être une boucle.
- Exemple des répétitions imbriquées :
  - Colonne de billes
  - Parcours en lacets

# Alternative et conditionnelle

- Le concept d'alternative ou de conditionnelle est à introduire dès que l'on souhaite programmer un calcul/comportement dont la suite devra dépendre d'informations acquises dans l'environnement pendant l'exécution du programme ou du résultat obtenu par un calcul intermédiaire.
- L'alternative est le concept permettant au programmeur de préciser par avance les actions que la machine devra effectuer dans différents états identifiés à l'avance.

# Alternative et conditionnelle – repères de progressivité

- Instruction conditionnelle ou "optionnelle"  
Forme : si condition alors ...
- Alternative. Forme : si condition alors... sinon ...
- La complexité des situations à programmer dépend du nombre de cas distincts à traiter et de la possibilité ou non de les résoudre partiellement en commun.
- L'imbrication de plusieurs alternatives permet de traiter un nombre arbitrairement grand de cas différents dans un programme.

# Les variables

- Le concept de variable peut se construire d'abord pour généraliser la programmation de calculs ou comportements analogues. Il est alors lié à la compétence de généralisation.  
Exemple : passer du dessin d'un carré de côté 100 à un carré de côté  $n$ .
- L'émergence de la variable est aussi liée à la compétence de "décomposition" d'un problème en sous-problèmes, les résultats intermédiaires devant être mémorisés dans des variables.
- Dans le sens le plus général, la variable est nécessaire pour mémoriser l'état d'exécution d'un programme. Le concept de variable est alors nécessité par le fait de devoir "mémoriser" des informations sur le comportement passé d'un programme, dont dépend l'exécution future de ce programme.

# Les variables – repères de progressivité

- 1) La variable "constante". Situation type : programme qui demande à son utilisateur la taille ou le nombre de côtés du polygone à dessiner. Correspond à la notion de constante ou de paramètre en mathématiques.
- 2) La variable à "affectation unique". Situation type : calcul défini par des formules permettant de calculer différentes grandeurs. Correspond à la notion de variable en mathématiques.
- 3) La variable au sens général : "emplacement de mémorisation" nommé, de façon explicite, dont la valeur peut être modifiée tout au long de l'exécution du programme. Exemple : « Retour au port en Scratch »

# Événements, et activités

- **Événement** : Phénomène qui arrive à un temps déterminé.
- **Activité** : Partie de programme pouvant être arrêtée ou en cours d'exécution.
- Une activité peut être déclenchée par l'occurrence d'un événement.
- Repères de progressivité :
  - 1) Événement provoqué par l'utilisateur
  - 2) Événement programmé

# Décomposer un programme

- La décomposition d'un problème complexe en sous-problèmes plus simples est une des compétences fondamentales de la programmation informatique.
- C'est aussi une manière de réutiliser une partie de la solution plusieurs fois, en ne l'ayant programmé qu'une fois.
- Isoler une partie de programme qui pourra resservir, nécessite de savoir "généraliser" pour identifier un traitement générique pouvant être effectué plusieurs fois, éventuellement avec un paramétrage différent à chaque fois.

# Décomposer – repères de progressivité

- La démarche de décomposition est un préalable à l'introduction de la notion de sous-programme (sans paramètres).
- L'introduction de la notion de procédure, sans, puis avec paramètres, peut être effectuée en fournissant d'abord des procédures déjà écrites pour en expérimenter l'utilisation.
- La création de procédures avec paramètres est une tâche plus complexe.

# Scripts et programmation parallèle

- Exécution séquentielle : "par construction" au plus un script s'exécute à un moment donné
- Exécution parallèle sans synchronisation :  
situations où le programmeur peut planifier indépendamment le comportement de chaque script
- Exécution parallèle synchronisée : hors programme cycle 4  
Situation d'accès concurrent à une variable.  
Plusieurs scripts déclenchés par le même événement.

# Repères de progressivité sur le curriculum

- Séquence
- Boucles
- Evénements/activités
- Variables
- Conditionnelles
- Décomposition
- Parallélisme
- Généralisation

A discuter !

# Perspectives

- Publication d'une brochure IREM développant ces éléments.
- Elaboration d'une proposition de progression conjointe en classe de mathématiques et de technologie avec des repères de progressivité par niveau : 5eme, 4eme, 3eme.
- Le groupe IREM « algorithmes et programmation au cycle 4 » se réunit 6 à 8 fois par an les mercredis après-midi et est ouvert aux collègues volontaires.