Algorithmes et programmation

1 Propositions de parcours

1. Une activité débranchée
2. 5P : 3 x 2 périodes prises sur la FMTTN et les Math.
   Tracer des polygones réguliers avec Scratch
3. 5-6P : 2 x 2 périodes prises sur la FMTTN et les Math.
   Un petit jeu vidéo avec Scratch
4. 6P : 2 x 2 périodes prises sur la FMTTN et les Math.
   Une version plus avancée de Tracer des polygones réguliers avec Scratch .

L'activité 4. Serait utile pour une deuxième année (6P) en guise de révision et comprend des techniques de programmation supplémentaires.

Les attendus du référentiel repris dans le programme sont rencontrés, y compris l'utilisation des logigrammes.

Des savoir-faire mathématiques seront exercés dans la foulée :

- Construire/dessiner des solides et figures
- Situer – placer/déplacer un objet dans un plan orthonormé
- Raisonnements logiques, conditions
- L'utilisation de variables

1.1 Activité déconnectée

1 période

En fonction du matériel ou des locaux dont dispose l'instit, on utilise n'importe quel damier sur lequel on place des objets qui représentent des obstacles et des couloirs par lesquels le robot pourra circuler pour aller d'un point à l'autre.

On peut donc travailler en « live » avec des élèves qui se déplacent sur des dalles ou grands carrelages ou avec des pions sur des damiers de jeux d'échecs.

Ce genre d'activité peut aussi être réalisée avec ce genre de matériel conçu pour les plus petits :

- Le robot BeeBot
- Le robot Botley ou la souris Code&Go de Learning Ressources

Déroulement de l'activité

J.P. Bihin, octobre 2024

	Par exemple
1. Définir l'objectif, ce que le robot devra réaliser.	Le robot doit se déplacer de la case « départ » à la case « arrivée » sans heurter d'obstacle et en passant par la case « X »
2. Distribuer les tâches et les rôles :	• L'analyste • Le programmeur • Le robot
3. Préciser les instructions que le processeur du robot est capable de comprendre. On suppose que l'ordinateur ne comprend qu'un petit nombre d'instructions en anglais (la plupart des langages de programmation sont en anglais) Ces instructions en anglais pourraient aussi être remplacées par des symboles . 	- Le programme est lancé quand on appuie sur un bouton « Start » : **START** - Avancer de … pas : **MOVE … STEPS** - Tourner de 90° à droite : **TURN RIGHT** - Tourne de 90° à gauche : **TURN LEFT** - Répéter … fois : **REPEAT … x** - Fin : **STOP**
4. L'analyste reçoit la demande et représente l'algorithme sous forme d'une séquence d'instructions notées en français dans des rectangles: **un logigramme**.	Logigramme pour un déplacement carré. 

J.P. Bihin, octobre 2024

5. Le programmeur utilise les instructions définies en 2 et retranscrit le **programme**. C'est l'étape qu'on appelle le codage. Il transmet ces instructions au robot. Remarque : la plupart des langages de programmation utilisent des termes anglais. Cela a du sens de faire apprendre ces quelques termes aux élèves.	1: START 2: MOVE 3 STEPS 3: TURN RIGHT 4: MOVE 3 STEPS 5: TURN RIGHT 6: MOVE 3 STEPS 7: TURN RIGHT 8: MOVE 3 STEPS 9 : TURN RIGHT 10 : STOP
6. Le même programme avec une boucle.	**Avec boucle** 1: START 2: REPEAT 4 x 3: MOVE 3 STEPS 4: TURN RIGHT 5: STOP
7. Le robot exécute à la lettre les instructions.	

Réaliser deux trajectoires. Et pour terminer, demander à l'analyste et au programmeur de préparer un algorithme qui représente un tracé carré de 3 pas de côtés (par exemple).

Le robot est envoyé dans un autre local ou se bouche les oreilles. Il reçoit le programme noté sur une feuille et l'exécute.

Vous souhaitez travailler davantage avec des activités débranchées ?

https://cdn.reseau-canope.fr/archivage/valid/contenus-associes-activites-pedagogiques-N-18069-29384.pdf

https://www.educode.fr/lordinateur-humain/

Remarque à propos des logigrammes

Le référentiel insiste beaucoup sur les logigrammes. Un peu trop, à mon avis.... Et notamment parce que les langages de programmation utilisés actuellement n'ont plus rien à voir avec les processus linéaires, constitués d'instructions qui s'exécutaient l'une après l'autre, comme on enfile des perles sur un fil. Aujourd'hui, la programmation est « orientée objet » et fait appel à des événements capables de lancer plusieurs processus

simultanément. Les logigrammes seront utiles pour des petites portions de codes, tout au plus.

Par contre, ils pourront trouver très utilement leur place dans d'autres domaines ou exercices techniques. N'hésitez pas à y faire appel pour représenter un processus de fabrication ou la réalisation d'une recette de cuisine.

1.2 Introduction à l'utilisation de Scratch et des cartes de programmation

Durée : une période.

Modalités pratiques :

- Pour ce premier exercice, **il n'est pas nécessaire de créer un compte Scratch** pour les élèves. Il faudra en revanche leur montrer comment renommer et enregistrer leur projet sur l'ordinateur.
- Idéalement **un pc et un jeu de carte par élève**. Un PC et un jeu de cartes pour deux peuvent aussi faire l'affaire.
- Les premiers projets étant enregistrés (et non dans le Cloud), veiller à ce que les élèves utilisent **le même ordinateur pendant toute la séquence**.

On présume qu'il s'agit d'un premier contact avec Scratch pour les élèves. Si ce n'est pas le cas, passer directement à l'étape suivante.

Présentation de Scratch :

- L'enseignant.e présente l'application et fait quelques démonstrations très simples. L'enseignant laisse ensuite un peu de temps aux élèves pour prendre la main et faire quelques essais. Le mieux est de leur proposer des petits défis très simples :
    - Place le sprite sur le bord gauche. Trouve et teste un bloc avec lequel le sprite va traverser en deux secondes la scène de gauche à droite.
    - Trouve et teste un bloc qui change la couleur de ton sprite.
    - Trouve et teste un bloc qui…
- Utiliser le tuto « Prise en main » et si on a du temps, éventuellement laisser les élèves suivre le tuto « Anime un nom »
  https://scratch.mit.edu/ideas

Utilisation des cartes de programmation

Expliquer le principe :

- Une carte défi et au verso, les consignes.
- On peut créer, proposer des variantes à une condition : avoir a fini tout le jeu de cartes.

1.3 5P Dessine des polygones

Quelques remarques préliminaires

Cette séquence couvre bien **plus que les attendus du référentiel**, et c'est voulu. Pour couvrir les attendus du référentiel, on pourrait s'arrêter à la carte 8. Je déconseille formellement de s'arrêter là. Il faut laisser les élèves aller le plus loin possible et essayer des variantes au programme, jouer avec les formes et les couleurs. Et pour cela, amener tous ceux qui le pourront jusqu'aux dernières cartes.

Profitons des exercices de programmation pour **exercer aussi des compétences mathématiques** et donc donner du sens aux notions vues dans ce cours, voire à celles qui seront vues plus tard. Ici, les élèves vont apprendre en pratiquant comme le disent les anglo-saxons : « Learning by doing ».

«Apprendre en faisant » veut aussi dire que l'enseignant doit accepter que **l'élève ne va pas tout comprendre tout de suite**. Les élèves doivent apprendre un maximum par eux-mêmes et en expérimentant.

1.3.1 Cartes 3 & 4 : prépare un sprite et un arrière-plan

RAS

1.3.2 Cartes 5 & 6 : Quelques réglages

Ici, on s'intéresse à deux propriétés du sprite : sa position sur la scène que l'ont peut modifier en la déplaçant avec la souris ou en modifiant la valeur de x et y dans la fenêtre des **propriétés**.

Le terme **propriété** est important en programmation. Dans ce cas, les propriétés d'un objet (= sprite) sont :

- son nom
- sa position horizontale (x = abscisse) et verticale (y = ordonnée)
- sa visibilité (oui/non)
- sa taille en % de la taille du costume
- sa direction en degrés

Le jeune se familiarisera avec ces propriétés en pratiquant.

1.3.3 Cartes 7 & 8 : Beetle effectue une trajectoire carrée

On entre dans le vif su sujet. L'élève essaie d'assembler les blocs proposés, teste, duplique et doit obtenir la trajectoire carrée et un retour à la position de départ.

Une petite procédure est suggérée pour **l'initialisation**. Il est important que l'élève comprenne que dans un programme, il faut toujours commencer par positionner l'objet

au point de départ de l'action. Ici, très concrètement, cela veut dire remettre l'objet au centre de la scène et le tourner dans la bonne direction.

1.3.4 Cartes 9 & 10 : deux défis Moins d'instructions SVP ! et Laisse une trace

On invite l'élève à utiliser une boucle avec le bloc *répéter… fois* , c'est-à-dire une instruction absolument élémentaire et facile à intégrer en programmation.

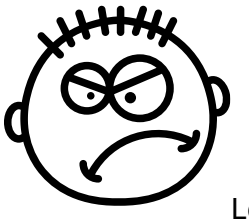
Dans un deuxième temps, ajouter une nouvelle famille de blocs très utile : l'extension *Stylo*. Simple comme bonjour !

1.3.5 Cartes 11 & 12 : un triangle

Il s'agit d'une étape importante. À ce stade, il faut laisser le temps aux élèves de réfléchir à la question suivante. Comment adapter le programme qui dessine un carré de façon à obtenir un triangle ? Il n'y a en principe aucun bloc à ajouter ou à retirer. Il faut juste modifier certaines valeurs : le nombre de répétition (= nombre de côtés du polygone) et le nombre de degrés de rotation.

Un raisonnement mathématique est esquissé sur la carte 11. C'est la notion d'angle externe dans un polygone et est utilisée ici. Une esquisse est proposée sur la carte 12. Le principe que l'élève devra retenir est : la somme des angles externes dans un polygone régulier est toujours égale à 360°.

Un petit coup de gueule



Le référentiel prévoit de faire intervenir les boucles en 6P. C'est totalement insensé. On ne ferait pas mieux avec un apprenti maçon à qui on donnerait comme instruction : « charge un brique dans ta brouette, amène-la jusqu'au lieu de la mise en œuvre et puis recommence l'opération dix fois » au lieu de « charge 10 briques dans la brouette et amène-la au lieu de la mise en œuvre».

L'utilisation des boucles, mais aussi des variables et des conditions doit intervenir le plus tôt possible dans l'apprentissage de la programmation. Sans elles, l'apprentissage de la programmation n'a aucun intérêt.

1.3.6 Cartes 13 & 14 : des rosaces

Ici aussi, on laissera du temps aux élèves pour multiplier les polygones et dessiner des rosaces, jouer avec les couleurs, déplacer les figures etc…

Dans l'exemple proposé, on reproduit 10 octogones en pivotant à chaque occurrence de 360/10° pour obtenir une rosace complète.