

Séquence

« Programmation créative avec Scratch Junior »



Auteur: Olivier Gagnac
Conseiller pédagogique DSDEN de la Dordogne
septembre 2016

Remerciements à Camille Prouillac, Assistante d'éducation à l'école des sciences de Bergerac pour tout le travail de tournage, mise en forme et montage des vidéos.
Remerciements à Vanessa Guionie, Sabrina Raymondau, Mylène Briand, David Philippe, Jean-Paul Cros, Gaelle Pestourie et Géraldine Aussat pour leur contribution et leur travail en classe, et à tous les relecteurs de la séquence.



Ce document est distribué sous licence libre Creative Commons CC-BY

Intentions pédagogiques

Aborder l'informatique via la notion de « pensée informatique » nous amène à mettre au cœur de la réflexion et de l'enseignement l'algorithmique et la programmation, dans une approche orientée « informatique créative ». L'entrée que nous privilégions ici consiste à partir à la découverte d'une interface de programmation avec « Scratch junior » sur tablette ou « Scratch » sur ordinateur, pour programmer une histoire écrite en classe, à partir d'un cahier des charges comme pour « la nuit des insomniaques »

<https://sites.google.com/site/nuitdesinsomniaques2015/>

La fiche 1, « Ecrire une histoire (langage naturel) », en annexe, propose un projet d'écriture qui permet de faire travailler les classes sur des contraintes communes, d'échanger les histoires produites et de comparer les transcriptions en langage de programmation. Ce travail d'écriture pourra se faire parallèlement aux premières séances de découverte du langage de programmation Scratch Junior.

En s'inspirant des pistes suggérées par Laurent Chéno, IGEN, les modalités d'apprentissage seront variées : individuel ou en groupe, en salle informatique ou en salle banale, travail en mode débranché, c'est-à-dire sans utilisation d'un dispositif informatique, sur tablette ou sur ordinateur, avec robot.

L'apprentissage associera trois types essentiels de démarche :

- une démarche de projet active et collaborative qui suppose la négociation des objectifs et la répartition des tâches, la communication entre élèves contributeurs d'un même projet et qui permet l'intervention de plusieurs disciplines ;
- une démarche de création : l'enseignement doit se traduire par la réalisation de productions collectives de programmes, au cours desquelles les élèves développent leur autonomie, mais aussi le sens du travail collaboratif ;
- une démarche interdisciplinaire qui favorise la mise en œuvre de diverses activités de création numérique.

L'apprentissage de la programmation se développera suivant une progression liée à des concepts de programmation plutôt qu'à des éléments de syntaxe du langage :

- la programmation événementielle : conception de séquences d'instructions déclenchées par un événement (appui de touche, son reçu par le microphone, motif « touché » par un personnage, etc.) ;
- l'initiation à la programmation parallèle : déclenchement par le même événement de deux ou plusieurs séquences d'instructions"

« La définition à laquelle font référence les promoteurs de l'enseignement de la pensée informatique à l'aide du langage Scratch est intéressante à double titre. D'une part, elle détaille les choses. D'autre part, elle permet de comprendre la vision sous-jacente aux ressources pédagogiques proposées autour de Scratch. Cette définition est :

§ *« [La pensée informatique] implique d'appréhender le monde selon l'approche employée en programmation par les développeurs de logiciels.*

§ *Cette approche peut être scindée en cinq grandes catégories :*

- *appréhender un problème et sa solution à différents niveaux (abstraction) ;*
- *réfléchir aux tâches à accomplir sous forme d'une série d'étapes (algorithmes) ;*
- *comprendre que pour résoudre un problème complexe il faut le décomposer en plusieurs problèmes simples (décomposition) ;*
- *comprendre qu'il est probable qu'un nouveau problème soit lié à d'autres problèmes déjà résolus par l'élève (reconnaissance de formes, ...)*
- *réaliser que la solution à un problème peut servir à résoudre tout un éventail de problèmes semblables (généralisation) ».*

La pensée informatique ne se réduit donc pas à l'algorithmique mais fait référence à une façon d'aborder les problèmes qui va en général conduire à l'écriture d'un algorithme en langage naturel plus ou moins structuré et, éventuellement, d'un programme, i.e., une traduction de l'algorithme dans un langage exécutable par une machine. »

Pierre Tchounikine, Professeur à l'Université Grenoble Alpes, Initier les élèves à la pensée informatique et à la programmation avec Scratch.



Éléments de contexte : références au programme et au socle commun

COMPÉTENCES TRAVAILLÉES	DOMAINES DU SOCLE
<p>Pratiquer des démarches scientifiques et technologiques</p> <p>Proposer, avec l'aide du professeur, une démarche pour résoudre un problème ou répondre à une question de nature scientifique ou technologique :</p> <ul style="list-style-type: none"> • Formuler une question ou une problématique scientifique ou technologique simple • Proposer une ou des hypothèses • Proposer une ou des expériences simples pour les tester • Interpréter un résultat, en tirer une conclusion 	<p>Domaine 4 : Les systèmes naturels et les systèmes techniques.</p> <ul style="list-style-type: none"> • Capacité à résoudre des problèmes
<p>Concevoir, créer, réaliser</p> <ul style="list-style-type: none"> • Identifier les évolutions des besoins et des objets techniques dans leur contexte • Décrire le fonctionnement d'objets techniques, leurs fonctions • Réaliser en équipe tout ou partie d'un objet technique répondant à un besoin 	<p>Domaine 5 : Les représentations du monde et de l'activité humaine</p> <ul style="list-style-type: none"> • Compréhension des sociétés dans le temps et dans l'espace
<p>Pratiquer des langages</p> <ul style="list-style-type: none"> • Rendre compte des observations, expériences, hypothèses, conclusions en utilisant un vocabulaire précis • Expliquer un phénomène à l'oral et à l'écrit • Participer à des échanges dans des situations diversifiées. • Adopter une attitude critique par rapport au langage produit. <p>Produire des écrits variés.</p>	<p>Domaine 1 : Les langages pour penser et communiquer.</p> <ul style="list-style-type: none"> • Comprendre, s'exprimer en utilisant la langue française à l'oral et à l'écrit • Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques
<p>S'approprier des outils et des méthodes</p> <ul style="list-style-type: none"> • Organiser seul ou en groupe un espace de réalisation expérimentale. • Utiliser les outils mathématiques adaptés 	<p>Domaine 2 : Les méthodes et outils pour apprendre.</p> <ul style="list-style-type: none"> • Outils numériques • Conduite de projets individuels et collectifs

ATTENDUS DE FIN DE CYCLE			
Matériaux et objets techniques		Espace et géométrie	Français
Repérer et comprendre la communication et la gestion de l'information.	Décrire le fonctionnement d'objets techniques, leurs fonctions et leurs constitutions	(Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations	- Recourir à l'écriture pour réfléchir et pour apprendre - Produire des écrits variés en s'appropriant les différentes dimensions de l'activité d'écriture
Connaissance et compétences associées			
Les élèves apprennent à connaître l'organisation d'un environnement numérique. Ils décrivent un système technique par ses composants et leurs relations. Ils découvrent l'algorithme en utilisant des applications visuelles et ludiques.	L'objet technique est à aborder en termes de description, de fonctions, de constitution afin de répondre aux questions : à quoi cela sert-il ? De quoi est-ce constitué ? Comment cela fonctionne-t-il ?	Se repérer, décrire ou exécuter des déplacements, sur un plan ou sur une carte. Accomplir, décrire, coder des déplacements dans des espaces familiers. Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran. - Vocabulaire permettant de définir des positions et des déplacements. - Divers modes de représentation de l'espace.	Ecrits de travail pour émettre des hypothèses, articuler des idées, hiérarchiser, lister. Ecrits réflexifs, pour expliquer une démarche, justifier une réponse. Connaissance des caractéristiques principales des différents genres d'écrits à produire. Mise en œuvre (guidée, puis autonome) d'une démarche de production de textes.
Repères de progressivité cycle 3			
Une initiation à la programmation est faite à l'occasion notamment d'activités de repérage ou de déplacement (programmer les déplacements d'un robot ou ceux d'un personnage sur un écran), ou d'activités géométriques (construction de figures simples ou de figures composées de figures simples).			

Description de la séquence

Séances	Objectifs	Domaines travaillés	Durée	Annexes	Vidéo de classe
0 Ecrire un histoire	<ul style="list-style-type: none"> - écrire une histoire en langage naturel pour la programmer en langage Scratch Junior - respecter des consignes d'écriture 	Français : lecture, langage écrit	...	Fiche 1 élève, « écrire une histoire »	Ecrire une histoire pour la programmer 
<p>REMARQUES : Plusieurs scénarii sont possibles. Pour faire le lien entre langage naturel et langage de programmation, d'autres consignes peuvent être données, on peut proposer le début d'une histoire et demander aux élèves d'imaginer la suite, ...</p> <p>Une autre entrée consiste à partir de l'application (personnages, objets et paysages proposés) pour écrire une histoire. Pour un groupe d'élève en difficulté sur le langage écrit, nous avons testé la possibilité de démarrer par la dictée à l'adulte en parallèle à la programmation, les élèves ont programmé la suite de leur histoire et sont revenus à l'écriture en langage naturel de la fin de leur histoire...</p>					Programmer une histoire pour l'écrire 
1 Scratch Junior : à quoi ça sert, comment ça marche ? Partie 1 Déplacer un personnage sur un quadrillage Partie 2	<ul style="list-style-type: none"> - Découvrir l'application - Mettre en évidence les principales fonctions de l'application : interface graphique et espace de programmation - Programmer un déplacement - S'exercer 	<ul style="list-style-type: none"> - Français : langage oral - Mathématiques - Sciences et technologie 	45 min	- Fiche 1 enseignant, « tutoriel Scratch Junior »	Découverte de l'application  A quoi ça sert ?  Mise en commun 
	<ul style="list-style-type: none"> - Coder un déplacement - Décoder un déplacement - Opérer des calculs sur des coordonnées 	<ul style="list-style-type: none"> - Mathématiques - Français : langage oral 	60 min	<ul style="list-style-type: none"> - Fiche 2 élève « exercices » - Fiche 2 enseignant « correction » 	Exercice 1  Exercices 2, 3  Exercice 5 
2 Programmer avec Scratch Junior	<ul style="list-style-type: none"> - se familiariser avec l'interface de l'application - coder des instructions données en langage naturel en langage de programmation - assembler des instructions simples pour réaliser un programme 	<ul style="list-style-type: none"> - Mathématiques - Sciences et technologie - Français : lecture, langage oral, acquisition de lexique 	60 min	<ul style="list-style-type: none"> - Fiche 3 élève « défis » - Fiche 5 élèves « briques Scratch Junior » 	Défis 

<p>3 Jeu de Nim Partie 1</p>	<ul style="list-style-type: none"> - découvrir le jeu - s'approprier la règle et le problème - élaborer une stratégie 	<ul style="list-style-type: none"> - Français : langage oral, et écrit - Mathématiques 	<p>50 min</p>	<ul style="list-style-type: none"> - Fiche 4 élève « jeu de Nim » 	<p>Découverte</p>  <p>Elaboration d'une stratégie</p> 
<p>4 Jeu de Nim Partie 2</p>	<ul style="list-style-type: none"> - écrire un algorithme - l'utiliser pour programmer un camarade pour gagner au jeu de Nim 	<ul style="list-style-type: none"> - Français : langage oral, et écrit - Mathématiques 	<p>50 min</p>	<ul style="list-style-type: none"> - Fiche 4 élève « jeu de Nim » 	<p>Ecriture de l'algorithme</p> 
<p>5 Programmer une histoire partie 1</p>	<ul style="list-style-type: none"> - installer les personnages de l'histoire dans l'application - installer les décors de l'histoire dans l'application - commencer à programmer - partager les découvertes éventuelles dans l'application - enregistrer le projet 	<ul style="list-style-type: none"> - Sciences et technologie - Français : langage oral 	<p>45 min</p>	<ul style="list-style-type: none"> - Fiche 5 élèves « briques Scratch Junior » 	<p>8</p> 
<p>6 Programmer une histoire partie 2</p>	<ul style="list-style-type: none"> - programmer et coordonner les déplacements des personnages - programmer et coordonner les dialogues des personnages - programmer les enchaînements entre les scènes - présenter les solutions techniques adoptées - enregistrer le projet - enregistrer le projet - analyser les programmes des autres groupes 	<ul style="list-style-type: none"> - Sciences et technologie - Français : langage oral - mathématiques 	<p>60 min</p>	<ul style="list-style-type: none"> - Fiche 5 élèves « briques Scratch Junior » 	<p>9</p> 
<p>7 Programmer une histoire partie 3</p>	<ul style="list-style-type: none"> - mettre au point les programmes - présenter les programmes - enregistrer le projet - analyser les programmes des autres groupes 	<ul style="list-style-type: none"> - Mathématiques - Sciences et technologie - Français : langage oral 	<p>60 min</p>	<ul style="list-style-type: none"> - Fiche 5 élèves « briques Scratch Junior » 	<p>10</p> 

Pour en savoir plus : Au cours de cette séquence, les 4 ingrédients de la programmation (instruction, boucle, instruction conditionnelle et variable avec le jeu de Nim), ainsi que 3 des 4 concepts informatiques sont abordés en rapport avec les activités proposées, langage, algorithme et machine. Pour en savoir plus, trois fiches documentaires :

- Qui a inventé les premiers algorithmes ? (lecture + résolution de problèmes)
- Alan Turing et le décodage des codes secrets nazis (lecture + exercices de codage et décodage)
- Grace Hopper (lecture + exercices sur le codage informatique des informations)

Fiche 1 : Ecrire une histoire (langage naturel)

Si le thème et le genre de votre histoire sont libres, elle devra cependant obligatoirement contenir les éléments suivants, insérés le plus habilement possible dans le texte :

- 1) Un des épisodes aura comme décors le parc d'une grande ville.
- 2) Le verbe voyager sera employé dans une phrase.
- 3) Une boîte à lettre doit apparaître dans l'histoire.
- 4) Un personnage doit prononcer la phrase : « ce fut un beau décollage »
- 5) L'histoire devra s'achever par « Et ce n'était que le début, ... »

Écrivez votre texte avec la police Cambria 12 et enregistrez-le en format texte (docx, doc ou odt). Choisissez un interligne de 1,5 une marge de 2 cm minimum à gauche et à droite. N'oubliez pas le titre.

Fiche 1 : Ecrire une histoire (langage naturel)

Si le thème et le genre de votre histoire sont libres, elle devra cependant obligatoirement contenir les éléments suivants, insérés le plus habilement possible dans le texte :

- 1) Un des épisodes aura comme décors le parc d'une grande ville.
- 2) Le verbe voyager sera employé dans une phrase.
- 3) Une boîte à lettre doit apparaître dans l'histoire.
- 4) Un personnage doit prononcer la phrase : « ce fut un beau décollage »
- 5) L'histoire devra s'achever par « Et ce n'était que le début, ... »

Écrivez votre texte avec la police Cambria 12 et enregistrez-le en format texte (docx, doc ou odt). Choisissez un interligne de 1,5 une marge de 2 cm minimum à gauche et à droite. N'oubliez pas le titre.

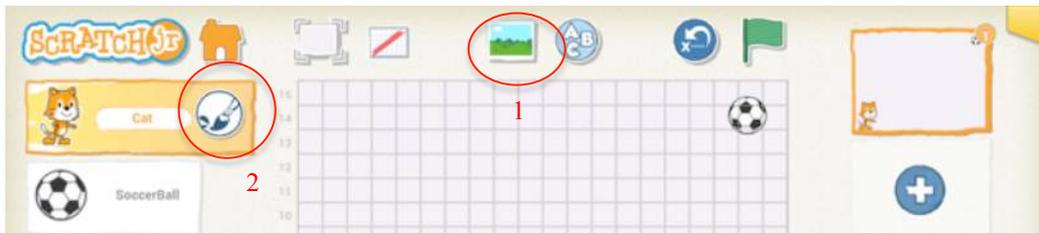
Fiche 1 : Ecrire une histoire (langage naturel)

Si le thème et le genre de votre histoire sont libres, elle devra cependant obligatoirement contenir les éléments suivants, insérés le plus habilement possible dans le texte :

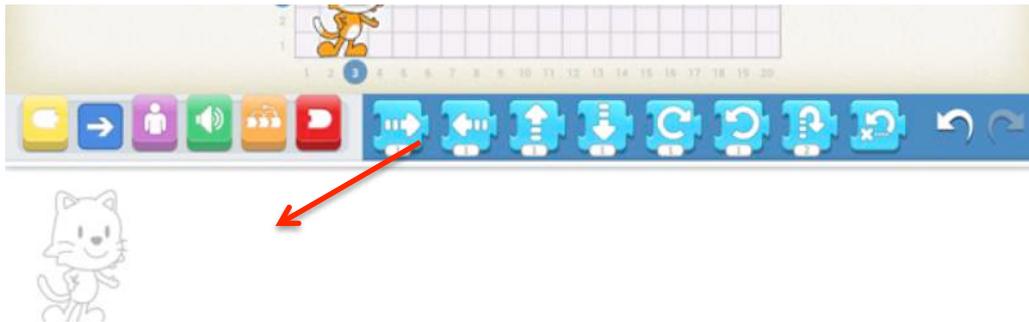
- 1) Un des épisodes aura comme décors le parc d'une grande ville.
- 2) Le verbe voyager sera employé dans une phrase.
- 3) Une boîte à lettre doit apparaître dans l'histoire.
- 4) Un personnage doit prononcer la phrase : « ce fut un beau décollage »
- 5) L'histoire devra s'achever par « Et ce n'était que le début, ... »

Écrivez votre texte avec la police Cambria 12 et enregistrez-le en format texte (docx, doc ou odt). Choisissez un interligne de 1,5 une marge de 2 cm minimum à gauche et à droite. N'oubliez pas le titre.





d'autres encore iront vers le langage de programmation (les briques de couleur) mais c'est la partie la moins intuitive, puisqu'il faut glisser ces briques dans l'espace réservé à l'écriture du code, en bas.



Phase 2 : Découvrir et utiliser les briques « Mouvement »

Retour en groupes et poursuite de la découverte orientée vers le langage de programmation, plus particulièrement les briques bleues (mouvement) qui permettent très rapidement de comprendre comment déplacer le personnage. Certains vont jusqu'à modifier l'opérateur en dessous des flèches et constatent que le personnage se déplace du nombre de cases correspondant à l'opérateur choisi, dans la direction de la flèche choisie.



Phase 3 : Programmer les personnage pour lui faire faire le tour de l'espace de programmation

Au bout de quelques minutes, on refait un point sur les différentes façons de faire bouger le personnage. Pour conclure la séance, on demande aux élèves d'écrire un programme permettant au personnage de faire le tour complet de l'espace de programmation en revenant au point de départ.

Phase 4 : Bilan

- A quoi ça sert? Scratch junior est une application permettant de créer des programmes.
- Comment ça marche? On donne des instructions à des personnages grâce à une série de boutons colorés, les bleus par exemple servent à les faire bouger.
- Questions/problèmes rencontrés? On peut attirer l'attention des élèves sur les différences de longueur des programmes proposés pour faire faire le tour de la scène au personnage, suivant les fonctionnalités découvertes (opérateurs, boucles).

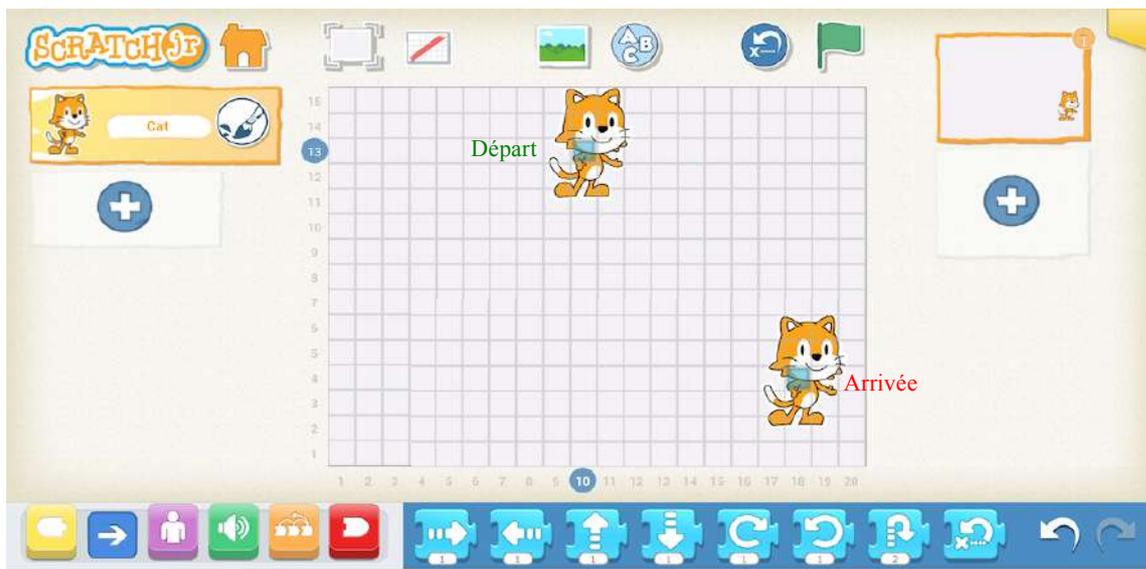
Phase 5 : Entraînement (fiche 2 : exercices)

Fiche 2 Exercices : Codage/décodage d'un déplacement sur un quadrillage.

1- Écrire un programme qui permet d'emmener Scratch (3 ; 3) jusqu'au ballon (19 ; 14) :



2- Tracez le trajet de Scratch (10 ; 13) au crayon sur le quadrillage, selon le programme codé ci-dessous, puis notez les coordonnées du point d'arrivée :



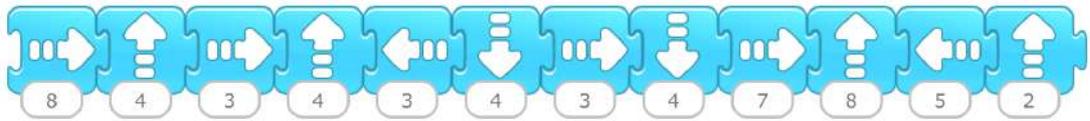
Coordonnées du point d'arrivée : (..... ;)



3- Parmi les programmes proposés, entourez ceux qui permettent à Scratch d'atteindre la porte de sa maison :



Code 1



Code 2



Code 3



4- Codez le déplacement de l'hippocampe vers l'étoile de mer en utilisant le moins de brique possible, sans toucher le plongeur et les poissons :



5. À partir du point de départ de coordonnées (10 ; 7), calcule les coordonnées des points d'arrivée à l'aide des lignes de codes proposées :

	<p style="text-align: right;">Départ</p> <p>(10 ; 7)</p> <p>(;)</p> <p style="text-align: right;">Arrivée</p>
	<p style="text-align: right;">Départ</p> <p>(10 ; 7)</p> <p>(;)</p> <p style="text-align: right;">Arrivée</p>
	<p style="text-align: right;">Départ</p> <p>(10 ; 7)</p> <p>(;)</p> <p style="text-align: right;">Arrivée</p>
	<p style="text-align: right;">Départ</p> <p>(10 ; 7)</p> <p>(;)</p> <p style="text-align: right;">Arrivée</p>

6. À partir du point de départ de coordonnées (12 ; 10), calcule les coordonnées des points d'arrivée à l'aide des lignes de codes proposées :



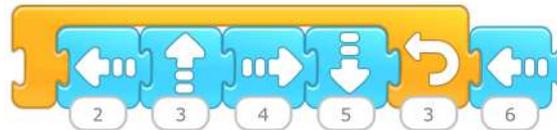
.....

.....



.....

.....



.....

.....

Séance 2 :

Objectif : Programmer un déplacement avec Scratch Junior

Concepts informatiques : Machine et langage

Phase 1 : Rappel de la séance précédente, insister sur les découvertes en rapport avec les briques bleues « mouvement », la possibilité de modifier la valeur des opérateurs, et éventuellement les boucles utilisées par certains groupes (briques « contrôle » orange).

Phase 2 : Correction en groupe des exercices de la fiche 1 à l'aide des tablettes puis mise en commun. Préciser que lors des vérifications avec les tablettes, il est inutile d'essayer de reproduire les paysages, il faut juste se concentrer sur les coordonnées des points de départ et d'arrivée.

Phase 3 : Défis (fiche 3)

Niveau 1

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3).
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée, vous ne devez pas utiliser de brique verte.
- Vous devrez utiliser le moins de briques possible.



Niveau 2

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3), en passant par le point de coordonnée (10 ; 5).
- Arrivé au point de coordonnée (10 ; 5), le chat devra marquer un temps d'arrêt, avant de sauter, puis de repartir.
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée, vous ne devez pas utiliser de brique verte.
- Vous devrez utiliser le moins de briques possible.



Niveau expert

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3), en passant par le point de coordonnée (10 ; 5).
- Jusqu'au point de coordonnée (10 ; 5), il devra changer de direction à chaque case.
- Arrivé au point de coordonnée (10 ; 5), le chat devra marquer un temps d'arrêt, avant de sauter sur place, puis de repartir.
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée et devra revenir au point de départ, coordonnée (2 ; 13).
- Vous devrez utiliser le moins de briques possible.



Phase 4 : Bilan

On peut donner des instructions à une machine en utilisant un langage spécial appelé langage de programmation. Les instructions sont rassemblées dans un programme pour que la machine les applique.

Fiche 3 Défis : Programmer

Niveau 1

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3).
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée, vous ne devez pas utiliser de brique verte.
- Vous devrez utiliser le moins de briques possible.

Niveau 2

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3), en passant par le point de coordonnée (10 ; 5).
- Arrivé au point de coordonnée (10 ; 5), le chat devra marquer un temps d'arrêt, avant de sauter, puis de repartir.
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée, vous ne devez pas utiliser de brique verte.
- Vous devrez utiliser le moins de briques possible.

Niveau expert

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3), en passant par le point de coordonnée (10 ; 5).
- Jusqu'au point de coordonnée (10 ; 5), il devra changer de direction à chaque case.
- Arrivé au point de coordonnée (10 ; 5), le chat devra marquer un temps d'arrêt, avant de sauter sur place, puis de repartir.
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée et devra revenir au point de départ, coordonnée (2 ; 13).
- Vous devrez utiliser le moins de briques possible.

Fiche 3 Défis : Programmer

Niveau 1

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3).
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée, vous ne devez pas utiliser de brique verte.
- Vous devrez utiliser le moins de briques possible.

Niveau 2

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3), en passant par le point de coordonnée (10 ; 5).
- Arrivé au point de coordonnée (10 ; 5), le chat devra marquer un temps d'arrêt, avant de sauter, puis de repartir.
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée, vous ne devez pas utiliser de brique verte.
- Vous devrez utiliser le moins de briques possible.

Niveau expert

- En utilisant le langage de programmation de Scratch Junior, déplacez le chat du point de coordonnée (2 ; 13) au point de coordonnée (19 ; 3), en passant par le point de coordonnée (10 ; 5).
- Jusqu'au point de coordonnée (10 ; 5), il devra changer de direction à chaque case.
- Arrivé au point de coordonnée (10 ; 5), le chat devra marquer un temps d'arrêt, avant de sauter sur place, puis de repartir.
- Le programme devra démarrer lorsqu'on touche le chat.
- Le chat vous remerciera à son arrivée et devra revenir au point de départ, coordonnée (2 ; 13).
- Vous devrez utiliser le moins de briques possible.



Séances 3 et 4:

- Objectif : Introduire la notion d'algorithme, sans ordinateur. Réinvestir les notions d'instructions, de langage et de programme, sans machine. Apprendre avec les mains.
- Concepts informatiques : algorithme, langage, instruction conditionnelle
- Matériel : 16 petits objets (jetons, clous, allumettes, . . . peu importe !), une fiche élève par élève à compléter avec les noms des joueurs, un feuille jaune pour noter les différentes étapes de la stratégie gagnante, fiche 4

Phase 1 : Rappel de la séance précédente. Insister sur le fait que les machines ne font qu'exécuter des instructions écrites dans un langage particulier, compréhensible par l'homme et la machine, sous forme de programme.

Phase 2 : Mise en situation à partir du jeu de Nim, 1^{er} tour.

L'approche débranchée à ce moment du module permet d'explicitier une procédure:

- appréhender un problème et sa solution à différents niveaux (**abstraction**) ;
- réfléchir aux tâches à accomplir sous forme d'une série d'étapes (**algorithmes**) ;
- comprendre que pour résoudre un problème complexe il faut le décomposer en plusieurs problèmes simples (**décomposition**) ;
- comprendre qu'il est probable qu'un nouveau problème soit lié à d'autres problèmes déjà résolus par l'élève (**reconnaissance de formes**, ...)
- réaliser que la solution à un problème peut servir à résoudre tout un éventail de problèmes semblables (**généralisation**)

La pensée informatique fait référence à une façon d'aborder les problèmes qui va conduire à l'écriture d'un algorithme en langage naturel plus ou moins structuré (et, éventuellement, d'un programme, i.e., une traduction de l'algorithme dans un langage exécutable par une machine).

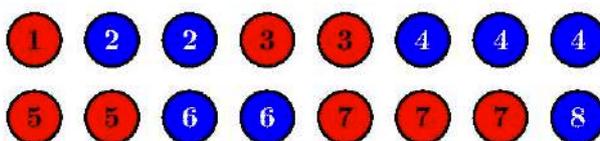
Consigne :

Chacun leur tour, les deux joueurs ramassent un, deux ou trois objets sur la table. Le joueur qui ramasse le dernier objet remporte la partie.

Le maître joue avec un élève et gagne à chaque fois...

Question :

Existe-t-il une stratégie qui permette de gagner à chaque fois ?



Le joueur bleu gagne

Les élèves sont mis en situation de jouer, par table, 4 joueurs, deux joueurs deux observateurs, avec pour objectif de découvrir une stratégie gagnante. (voir fiche 3 jeu de Nim)

Le rôle des observateurs est important car dégagés de la nécessité de jouer, ils découvrent assez rapidement que pour gagner, il faut laisser 4 jetons à l'adversaire à l'avant dernier coup. C'est un début de stratégie.

Par table de 4, les élèves font trois parties. A la fin, on fait un bilan pour savoir qui gagne à chaque fois ou le plus à chaque table. Les vainqueurs sont amenés à détailler les stratégies qu'ils utilisent.

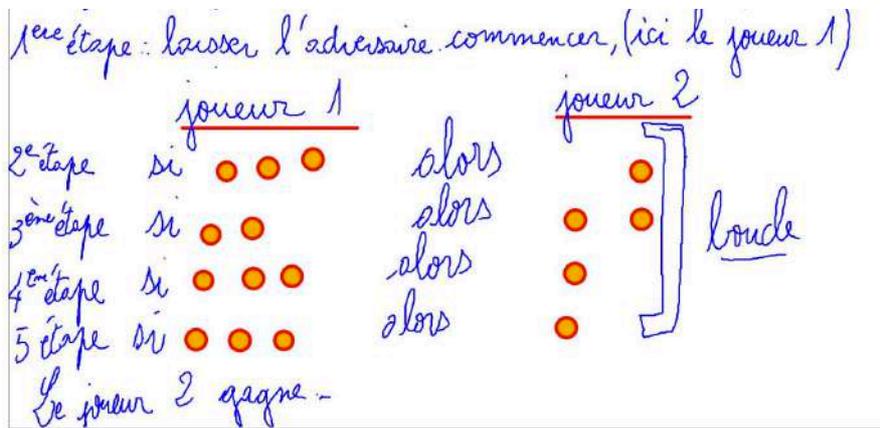
Phase 3: Mise en situation à partir du jeu de Nim, 2^{ème} tour.

Un deuxième tournoi s'engage, on peut regrouper les vainqueurs des différentes tables à la même table, on donne à toutes les tables la consigne de tester les stratégies proposées et de compléter la feuille jaune, avec un programme à rédiger par étape.

date 17/03/2016
 Le jeu de Nim
 Mon programme pour gagner:
 1^{ère} étape Laisser commencer, dès que
 4 jets sur huit
 2^{ème} étape Laisser commencer. si le j1
 prend 1 alors j2 prend 3
 3^{ème} étape si j1 prend 2 alors j2 en
 prend 2
 4^{ème} étape si le j1 prend 3 le j2 prend
 1
 5^{ème} étape si le j1 en prend 2 le j2 en prend
 2

date 17/03/16
 Le jeu de Nim
 Mon programme pour gagner:
 1^{ère} étape je ne prend rien si il en prend 1 je prend 3
 2^{ème} étape si il prend 3 jetons je prend 1 jeton
 3^{ème} étape si il en prend 2 jetons je prend
 2 jetons
 4^{ème} étape

A la fin, le maître choisit un élève qui lui donne des instructions pour réaliser le programme gagnant. Le programme proposé ici comporte 5 étapes, avec répétition d'instructions conditionnelles **SI...ALORS**.



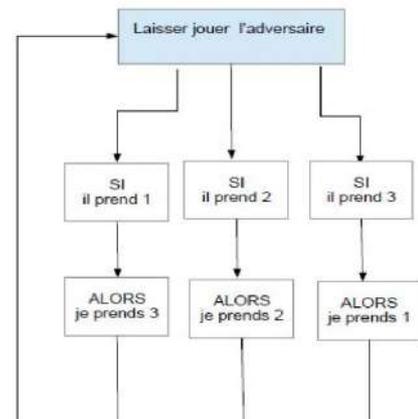
Phase 4: Comme pour les défis de la séance 2, il s'agit maintenant de réécrire ce programme pour ne pas avoir à répéter plusieurs fois la même instruction, en introduisant une boucle pour les étapes 2, 3, 4 et 5. Les élèves ont bien compris qu'il fallait que 4 jetons, au total, soient prélevés par les deux joueurs, à chaque tour. Les boucles, dans un programme, permettent de répéter certaines instructions plusieurs fois sans avoir à recoder plusieurs fois ces instructions.

Formulation d'élève :

1^{ère} étape. Laisser j1 commencer
 2^{ème} étape. Si j1 prend 1, 2, 3 jetons alors le j2 prend 1, 2, 3 pour que $j1 + j2 = 4$

On peut alors tendre vers une formulation du style « si J1 commence, alors J2 doit laisser sur la table un multiple de 4 » (12 puis 8 puis 4 jetons).

Le jeu de Nim est sans suspense : le premier à jouer perd, car il existe une astuce pour que le deuxième joueur gagne à tous les coups. La stratégie gagnante est de laisser 4, 8, 12 ou 16 objets à l'adversaire (un multiple de 4).



Pour se convaincre de l'efficacité de la stratégie gagnante, prenons le dernier tour comme exemple. Il reste 4 objets, et J1 joue :

- _ si J1 prend 1 objet, J2 en prend 3 (dont le dernier) ;
- _ si J1 prend 2 objets, J2 en prend 2 (dont le dernier) ;
- _ si J1 prend 3 objets, J2 en prend 1 (le dernier).

Dans ce cas, si J2 sait jouer, J1 perd à tous les coups. En appliquant la même méthode, J2 peut guider le jeu de manière à passer de 16 objets à 12, puis 8 et enfin 4. Donc, si J2 sait jouer, J1 a perdu la partie avant même de commencer.

L'intérêt de ce jeu, au delà de la facilité de sa mise en œuvre, réside dans le fait que l'algorithme dit quelle instruction effectuer quand une condition est vérifiée :

SI joueur 1 prend 1 jeton, **ALORS** joueur 2 prend 3 jetons.

Il permet l'introduction de ces instructions conditionnelles et du mode de raisonnement si...alors.

Phase 5 : Nim, le robot imbattable.

Tandis que certains élèves fournissent assez rapidement un algorithme permettant de gagner à chaque fois, d'autres ont du mal à trouver une stratégie à saisir la méthode proposée par leurs camarades lors des mises en commun.

On peut alors proposer aux « programmeurs » de trouver le moyen de « programmer » leurs camarades pour qu'ils gagnent aux aussi au jeu de Nim, leur demander de trouver le « langage de programmation » et le programme le plus adapté (court, long, avec ou sans boucle) pour aider leurs camarades.

Phase 6 : Bilan

En informatique comme pour le jeu de Nim, un algorithme est une méthode, un programme, qui permet de résoudre un problème.

Ici, il est rédigé en langage « naturel », le français et dit quelle instruction effectuer lorsqu'une condition est vérifiée : **SI**...**ALORS**.

Les boucles, dans un programme, permettent de répéter certaines instructions plusieurs fois sans avoir à recoder plusieurs fois ces instructions.

Phase 7 : Prolongements

Le « robot géomètre » peut être chargé de réaliser une figure géométrique à partir du programme proposé par un autre élève ou groupe d'élève. Les erreurs de programme peuvent servir à introduire la notion de bug. Plusieurs autres activités débranchées sont proposées en annexe.

Fiche 4 : Le jeu de Nim

Matériel : 16 petits objets (jetons, clous, allumettes, . . . peu importe !)

Consigne :

Chacun leur tour, les deux joueurs ramassent un, deux ou trois objets sur la table. Le joueur qui ramasse le dernier objet remporte la partie.

Existe-t-il une stratégie qui permette de gagner à chaque fois ?

Tableau de score (écris le nom sur les pointillés et entoure le vainqueur) :

Tour 1

joueur 1..... contre joueur 2.....
joueur 3..... contre joueur 4.....

Tour 2

joueur 1..... contre joueur 3.....
joueur 2..... contre joueur 4.....

Tour 3

joueur 1..... contre joueur 4.....
joueur 2..... contre joueur 3.....

Qui a le plus de victoires ? Quelle est la stratégie gagnante?

Explique la stratégie gagnante en écrivant un programme commençant par :

Pour gagner à chaque fois, il faut...

Etape 1 :

Etape 2 :

Etape 3 :

...

...

...



Séances 5, 6 et 7:

- Objectif : Utiliser Scratch Junior pour coder une histoire écrite en classe et réaliser un petit film d'animation
- Concepts informatiques : programmation
- Matériel : tablettes, tutoriel papier Scratch Junior et histoire écrite en classe

Phase 1 : Rappel des fonctionnalités de Scratch Junior découvertes lors des séances précédentes et lecture de l'histoire rédigée en classe.

Phase 2 : Consigne : créez, à l'aide de l'application, un petit film d'animation à partir de l'histoire que vous avez rédigée à partir des contraintes proposées dans la fiche 1 :

« Si le thème et le genre de votre histoire est libre, elle devra cependant obligatoirement contenir les éléments suivants, insérés le plus habilement possible dans le texte :

- 1) *Un des épisodes aura comme décors le parc d'une grande ville.*
- 2) *Le verbe voyager sera employé dans une phrase.*
- 3) *Une boîte à lettre doit apparaître dans l'histoire.*
- 4) *Un personnage doit prononcer la phrase : « ce fut un beau décollage »*
- 5) *L'histoire devra s'achever par « Et ce n'était que le début, ... »*

Écrivez votre texte avec la police Cambria 12 et enregistrez-le en format texte (docx, doc ou odt). Choisissez un interligne de 1,5 une marge de 2 cm minimum à gauche et à droite. N'oubliez pas le titre. »

Contraintes liées à l'application :

- il ne pourra pas y avoir sur une seule tablette plus de 4 scènes dans des décors différents
- vous utiliserez des personnages et des décors de la banque d'images fournies par l'application
- si le travail est réparti entre les différents groupes (une scène par groupe de la même histoire), et que vous souhaitez avoir le film en entier sur une tablette, il faudra coder l'ensemble des scènes sur au moins un tablette

Phase d'exploration des solutions graphiques offertes par l'application.

Phase 3 : Comparaison entre les groupes des solutions de codage imaginées, vérification de la cohérence des choix graphiques (personnages, taille et couleur des personnages et des décors, paysages,...) entre les groupes pour respecter la continuité narrative.

Phase 4 : Organisation du travail, définition des objectifs de chaque groupe, mise en route.

A partir de ce moment, le maître devient personne ressource, régule l'activité des groupes, ...

Des temps de présentation de l'état d'avancée des différents groupes sont régulièrement proposés, les solutions imaginées par certains peuvent aider les autres, ces temps permettent aussi aux groupes de se situer par rapport aux autres en terme d'avancement du travail.

Qui a inventé les premiers algorithmes ?

Pour comprendre l'origine de ce mot, il faut aller faire un tour vers l'an 800, en Perse... et si nous allions à la rencontre de monsieur "Algorithme" ?

Biographie

Al Khwarizmi est né vers 780, il est originaire de l'Ouzbékistan, il vécut à Bagdad en Irak pendant la dynastie des Abbassides, empire qui s'étendait de l'Afrique du Nord en passant par la Perse jusqu'en Inde et en Espagne (avant appelée Andalousie). Son nom est à l'origine du mot « algorithme » et le titre de l'un de ses ouvrages (*Abrégé du calcul par la restauration et la comparaison*) à l'origine du mot « algèbre ». L'utilisation des chiffres arabes et leur diffusion dans le Moyen-Orient et en Europe sont dues à un autre de ses livres qui traite des mathématiques indiennes.



A cette époque, le calife Abd Allah al Mahmoun fonda à Bagdad la Maison de la Sagesse (Bait Al-Hikmat), institut des sciences qui avait pour vocation de collecter les œuvres scientifiques et philosophiques des pays de la dynastie, et de les traduire en arabe afin qu'ils soient utilisés par les scientifiques pour leurs recherches. Al Khwarizmi, comme de nombreux penseurs, astronomes, mathématiciens, etc., travailla au sein de la Maison de la Sagesse.

Plus qu'un simple traducteur, Al Khwarizmi y développa des connaissances en géographie, astronomie et en algèbre à partir des textes qu'il traduisit du grec et de l'hindou. Al Khwarizmi publia son livre *Abrégé du calcul par la restauration et la comparaison* (Kitāb al-jabr wa al-muqābala) pour répondre à une demande du calife qui consistait à développer des instruments pour résoudre les problèmes de l'état comme par exemple le calcul des impôts, des héritages, les arpentages (calcul de surface, volume).

Al Khwarizmi posa les bases de l'algèbre ("al-jabr") et de l'algorithme (qui est la déformation latine de son nom) en instaurant la notation décimale (déjà utilisée par les hindous), en introduisant la notion d'inconnue (x), ...

Il présente des démonstrations mathématiques permettant d'arriver au résultat, intègre à ces démonstrations des additions, des soustractions, des multiplications et des divisions et des racines carrées. Il illustre ces exemples par des situations économiques (pour les héritages par exemple, ...).

En cliquant sur le lien ci-dessous, vous accèderez à une autre vidéo de présentation de l'œuvre d'Al Khwarizmi

- [La vidéo "Les équations" de la série "Petits contes mathématiques" sur le site Universciences](#)

Comme Al Khwarizmi, saurais-tu trouver une méthode pour résoudre les problèmes de logique suivants ?

(Difficulté : * = facile ; ** = moyen ; * = difficile)**

1. Les métiers *

Mr Peintre, Mr Maçon et Mr Menuisier sont 3 amis portant le même nom que leurs 3 métiers, mais pas nécessairement dans cet ordre.

Mr Menuisier n'est pas peintre.

Mr Maçon n'est pas menuisier.

Mr Menuisier est menuisier.

Mr Maçon n'est pas peintre.

Qui fait quoi ?

2. Les comédiens **

Antoine, Bruno, Charles, Denis et Etienne tiennent le rôle d'un roi, d'un soldat, d'un fou, d'un bourreau et d'un prisonnier dans une pièce de théâtre.

1. Antoine, Bruno et le prisonnier ne connaissent pas encore leur texte par cœur.

2. Pendant les pauses, le soldat joue aux cartes avec Denis.

3. Antoine, Bruno et Charles critiquent la manière de jouer du bourreau.

4. Le fou apprécie le jeu de Bruno, de Charles et d'Etienne, mais déteste celui du soldat.

Peux-tu attribuer à chacun d'eux son rôle ?

3. Les couples à la plage ***

1 La femme de Rémi fait une partie de pétanque avec le mari de Geneviève.

2 Barbara joue au ping-pong avec son mari Nicolas.

3 Jacques et sa femme Nadine font la sieste.

4 Isabelle n'est pas la femme de Thomas.

5 Geneviève fait une promenade avec Gisèle.

6 Robert n'est pas marié avec Geneviève.

Qui a épousé Robert ?

4. Les maisons ***

Attention, plus difficile ! On multiplie les dimensions. Chaque case du tableau a maintenant 5 possibilités.

5 maisons, 5 nationalités, 5 boissons, 5 produits du tabac, 5 animaux favoris.

15 indices pour tout démêler...

1. Le Britannique habite dans une maison rouge.

2. Le Suédois possède un chien.

3. Le Danois boit du thé.

4. La maison verte est immédiatement à gauche de la maison blanche.

5. Le propriétaire de la maison verte boit du café.

6. La personne qui fume des cigares de marque Pall Mall élève des oiseaux.

7. Le propriétaire de la maison jaune fume des Dunhill.

8. L'homme qui vit dans la maison du milieu boit du lait.

9. Le Norvégien habite dans la première maison.

10. L'homme qui fume des Blend demeure dans la maison située à côté de celui qui garde des chats.

11. L'homme qui possède des chevaux habite à côté de celui qui fume des Dunhill.

12. Le propriétaire qui fume des Gauloises boit de la bière.

13. L'Allemand fume des cigares de marque Prince.

14. Le Norvégien habite à côté de la maison bleue.

15. Le voisin de l'homme qui fume des Blend, boit de l'eau.

À qui appartient le poisson?



PROBLEMES DE LOGIQUE

Ces problèmes ne sont pas très difficiles à résoudre si on lit l'énoncé et si on procède avec la bonne méthode : On répond à ce genre de problème par un tableau à double entrée : Les indices vont permettre de valider (mettre un **O** comme **Oui** dans la bonne case) les possibilités ou de les éliminer (mettre une croix **X**).

On peut demander aux élèves de remplir le tableau par équipe : chaque équipe capable d'ajouter un **O** ou un **X** dans le tableau marque un point.

1. Les métiers

	Peintre	Maçon	Menuisier
M. Peintre	O	X	X
M. Maçon	X	O	X
M. Menuisier	X	X	O

2. Les comédiens

« joue le rôle de »	Roi	Soldat	Fou	Bourreau	Prisonnier
Antoine	X	O	X	X	X
Bruno	O	X	X	X	X
Charles	X	X	X	X	O
Denis	X	X	O	X	X
Etienne	X	X	X	O	X

3. Les couples à la plage

	Barbara	Isabelle	Geneviève	Gisèle	Nadine
Rémi	X	O	X	X	X
Nicolas	O	X	X	X	X
Jacques	X	X	X	X	O
Thomas	X	X	O	X	X
Robert	X	X	X	O	X

4. Les maisons

Le poisson appartient à l'Allemand:

jaune	bleue	rouge	verte	blanche
Norvégien	Danois	Britannique	Allemand	Suédois
Dunhill	Blend	Pall Mall	Prince	Gauloises
chats	chevaux	oiseaux	poisson	chien
eau	thé	lait	café	bière

Alan Turing et le décryptage des codes secrets nazis

10.05.2012, par
[Charline Zeitoun](#)

Mis à jour le 06.06.2014



Les Nazis codaient leurs messages avec des machines qui échangeaient les lettres tapées sur le clavier. Le modèle utilisé par la marine fut craqué dès 1942 par les Anglais avec l'aide d'Alan Turing.

Le débarquement du 6 juin 1944 n'aurait peut-être pas eu lieu sans le mathématicien Alan Turing. Celui-ci a en effet joué un rôle déterminant pour briser le code secret de la marine allemande. Explications avec François Morain, du Laboratoire d'informatique de l'École polytechnique.



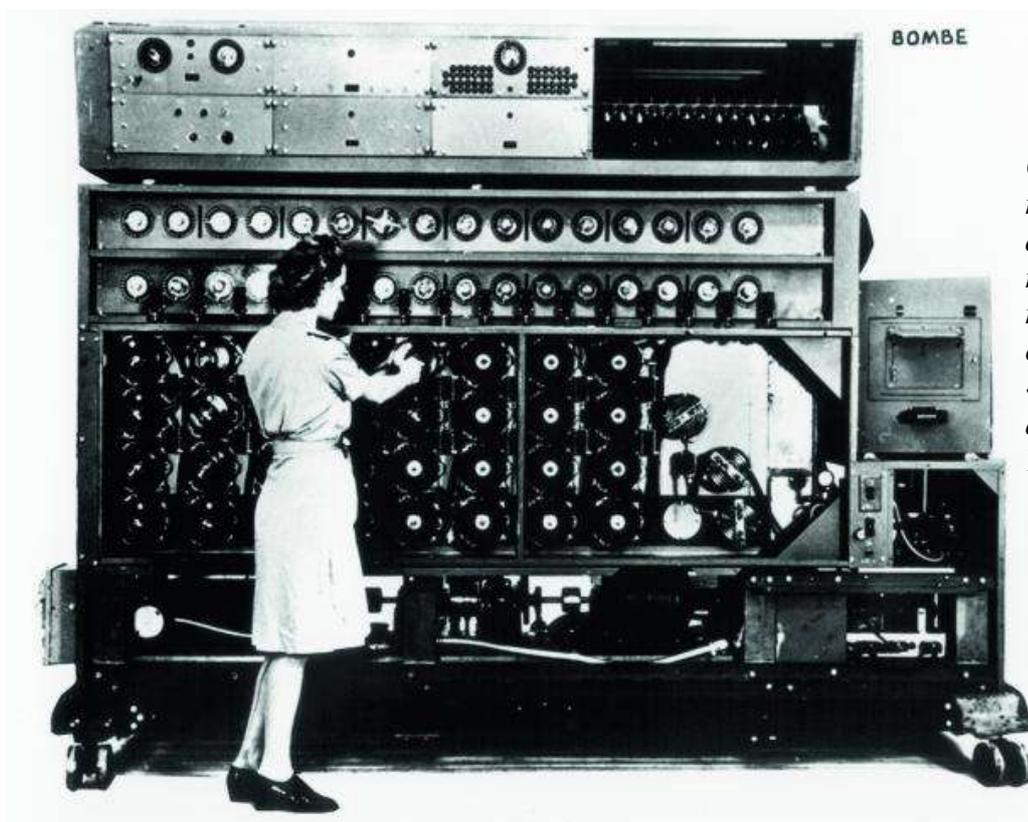
En septembre 1939, [Alan Turing](#) rejoint le manoir de Bletchley Park, quartier général des services de renseignement britannique, pour briser le système de cryptage des sous-marins allemands, réputé inviolable. Il y parvient avec ses collègues dès 1942, contribuant à éviter l'invasion de l'Angleterre.

Comment ont-ils fait ?

François Morain: Bletchley Park était une usine à décrypter dans laquelle la quasi-totalité des mathématiciens de Cambridge et d'Oxford s'étaient enrôlés, travaillant dans des baraquements et faisant les trois-huit. Les réflexions de Turing sur le calcul le plaçaient en première ligne pour participer à cette entreprise. La marine allemande utilisait une machine à crypter, Enigma. Il s'agissait d'une version ultrasophistiquée d'autres modèles allemands du même nom, employés par les armées de terre et de l'air, et déjà étudiés par les services de renseignement polonais. Ces derniers avaient en effet construit des "bombes", machines qui permettaient de tester toutes les clés de chiffrement possibles dans un minimum de temps. Faire vite était indispensable, car la clé changeait tous les jours.

Turing n'a-t-il fait qu'utiliser les bombes conçues et construites par les Polonais ?

F. M. : Non, les bombes des Polonais ne permettaient de casser que les premières versions d'Enigma. Turing a participé à la construction de bombes adaptées à la version utilisée par la marine. Surtout, il a exploité différentes faiblesses dans la façon dont les Allemands utilisaient leur système de codage. Ils commençaient, par exemple, leurs messages par des formules de politesse assez convenues, de type *Herr Kommandant*, faciles à deviner. Tout comme les messages très courts et stéréotypés qu'ils envoyaient régulièrement pour donner la météo ou annoncer qu'il ne se passait rien. En cryptanalyse, justement, on cherche souvent d'abord à deviner ce que des mots veulent dire et ensuite on teste si la clé de chiffrement ainsi définie fonctionne sur l'ensemble du ou des messages. C'est dans cette approche déductive que Turing a fait preuve d'une grande intuition pour réduire le nombre de combinaisons à tester. C'est là que résident la prouesse et l'intelligence.



Cette « bombe », machine pour décrypter les messages codés, est un modèle de l'US Navy construite d'après les « bombes » anglaises, avec l'aide d'Alan Turing.

Que peut-on dire de l'apport de Turing à la cryptographie ?

F. M. : Peu de choses en réalité. Ses travaux n'ont jamais fait l'objet de publications scientifiques. On ne dispose que de son carnet de laboratoire, inaccessible jusqu'en 1996 à cause du secret militaire, et qui n'a pas été rédigé de manière scientifique. Il y consigne tout ce qu'il sait, mais on ignore la part entre ses propres réflexions et les informations glanées auprès des autres scientifiques et agents des services secrets. Ensuite, il faut bien comprendre la distinction entre la cryptographie, science du chiffrement qui nécessite la production d'algorithmes et de théories, et la cryptanalyse, qui consiste à casser un code secret. Il n'y a pas de théorie du cassage. Il s'agit essentiellement d'avoir l'intelligence et l'astuce d'exploiter les faiblesses d'un système ou de son utilisation par les opérateurs humains. Quoi qu'il en soit, la façon dont Turing y est parvenu à l'époque, sans ordinateur, reste un immense tour de force et fait partie de la mythologie de la discipline.

1- A ton tour, comme Alan Turing et son équipe, essaye de décrypter les messages secrets suivants.

Message 1 :

14	15	21	19		1	22	15	14	19				
3	15	13	13	5	14	3	5		1				
16	18	15	7	18	1	13	13	5	18				
14	15	20	18	5		8	9	19	20	15	9	18	5
19	21	18		20	1	2	12	5	20	20	5		

Quelle est la clé de cryptage utilisée ?

Message 2 :

T	D	S	B	U	D	I		K	V	O	J	P	S
F	T	U		V	O	F							
B	Q	Q	M	J	D	B	U	J	P	O			
Q	P	V	S		U	B	C	M	F	U	U	F	

Quelle est la clé de cryptage utilisée ?

2- Trouve une nouvelle clé de cryptage pour coder le message suivant :

Thymio est un robot programmable.

T	H	Y	M	I	O		E	S	T		U	N	
R	O	B	O	T									
P	R	O	G	R	A	M	M	A	B	L	E		



1- A ton tour, comme Alan Turing et son équipe, essaye de décrypter les messages secrets suivants. **CORRECTION**

Message 1 :

14	15	21	19		1	22	15	14	19				
N	O	U	S		A	V	O	N	S				
3	15	13	13	5	14	3	5		1				
C	O	M	M	E	N	C	E		A				
16	18	15	7	18	1	13	13	5	18				
P	R	O	G	R	A	M	M	E	R				
14	15	20	18	5		8	9	19	20	15	9	18	5
N	O	T	R	E		H	I	S	T	O	I	R	E
19	21	18		20	1	2	12	5	20	20	5		
S	U	R		T	A	B	L	E	T	T	E		

Quelle est la clé de cryptage utilisée ?

Les lettres sont remplacées par le nombre qui correspond à leur place dans l'alphabet.

Message 2 :

T	D	S	B	U	D	I		K	V	O	J	P	S
S	C	R	A	T	C	H		J	U	N	I	O	R
F	T	U		V	O	F							
E	S	T		U	N	E							
B	Q	Q	M	J	D	B	U	J	P	O			
A	P	P	L	I	C	A	T	I	O	N			
Q	P	V	S		U	B	C	M	F	U	U	F	
P	O	U	R		T	A	B	L	E	T	T	E	

Quelle est la clé de cryptage utilisée ?

Chaque lettre est remplacée par la lettre qui la suit dans l'alphabet.

2- Trouve une nouvelle clé de cryptage pour coder le message suivant :

Thymio est un robot programmable.

T	H	Y	M	I	O		E	S	T		U	N	
R	O	B	O	T									
P	R	O	G	R	A	M	M	A	B	L	E		



Grace Hopper

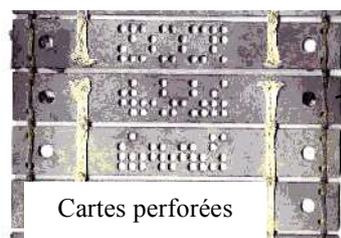
Grace Hopper est une mathématicienne Américaine, née en 1906. Elle grandit en montrant une véritable curiosité pour les sciences et la technologie (par exemple, en démontant les réveils de sa maison les uns après les autres, jusqu'à arriver à en remonter un entièrement... à l'âge de 7 ans). Ses parents encouragent son goût pour les sciences tout en veillant à valoriser ses choix plutôt que suivre la tradition en matière d'éducation des jeunes filles Américaines. Ainsi, Grace sera une des rares femmes de son époque à étudier à l'université de Yale et à obtenir un doctorat de mathématiques.



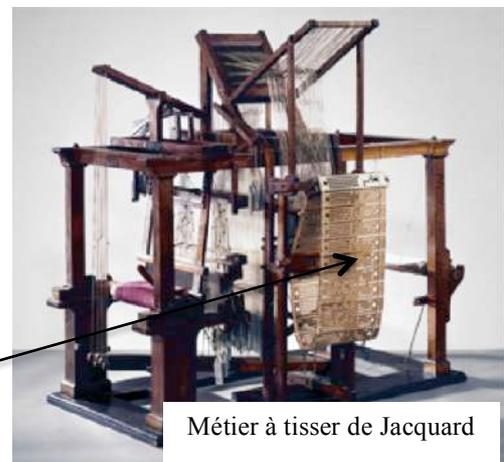
Ordinateur Mark 1

Pendant la deuxième guerre mondiale, elle entre dans la marine, où elle est nommée lieutenant, et rejoint l'équipe qui travaille sur le **Mark 1**, le premier ordinateur des États-Unis. Grace Hopper fait alors partie du premier groupe capable de programmer cette machine.

Cet ordinateur prend ses instructions et ses données à l'aide de **cartes perforées**, comme cela était prévu sur le métier à tisser de Jacquard environ deux siècles auparavant.



Cartes perforées



Métier à tisser de Jacquard

Comment programmer un ordinateur autrement qu'avec des cartes perforées ?

Dans les années 1950, Grace Hopper pense que l'informatique devrait être accessible à un plus grand nombre de personnes, et pas seulement aux spécialistes des ordinateurs et aux mathématiciens. Elle défend l'idée que les cartes perforées sont un frein au développement de l'informatique et travaille sur un projet de « **langage de programmation** » qui serait proche de l'anglais, et qui faciliterait l'écriture des programmes.

La difficulté est alors de concevoir le programme capable de transformer des instructions données dans un langage proche de l'anglais, facilement compréhensible par un **humain**, en instructions compréhensibles par la **machine**...

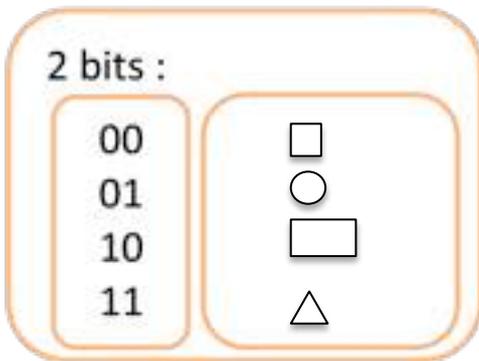
A toi de jouer, ...

Exercice1 « 1bit » : Si on code le noir par la valeur « 0 », et le blanc par la valeur « 1 », en coloriant la grille on obtient.....



1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1

Exercice2 « 2bits » : Avec 2 bits, on peut coder, par exemple, les 4 figures géométriques de base : Carré, Cercle, Rectangle, Triangle.

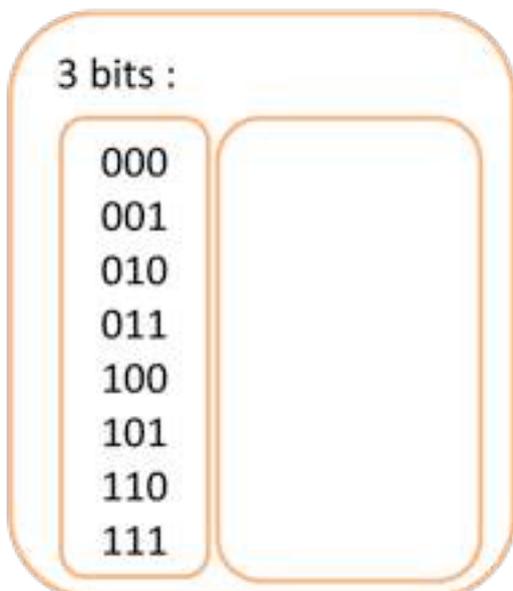


Code l'algorithme suivant en langage binaire.



○	□	△	▭	○	□	△
.....

Exercice 3 « 3bits » : Ecris un exemple de liste dont tous les éléments peuvent être codés sur 3 bits.



Exercice 4: Avec 1 bit, on peut coder une liste de deux éléments (noir et blanc par exemple dans l'exercice 1). Avec 2 bits, on peut coder une liste de 4 éléments, avec 3 bits une liste de 8 éléments. Combien faut-il de bits pour coder les éléments de la liste suivante : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, x, /, =, C.
Code chacun des ces éléments en langage binaire (avec des 1 et des 0).

Tu peux faire tes essais de combinaisons ci-dessous.

... bits :

	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	+
	-
	x
	/
	=
	C

Exercice 5, défi : détermine combien de bits il faut juxtaposer au minimum pour pouvoir coder chacune des 26 lettres de l'alphabet.

Exercice 6 : Table de correspondance pour coder chacune des 26 lettres de

5 bits	00000	00001	00010	00011	00100	00101	00110	00111
Caractère	A	B	C	D	E	F	G	H
5 bits	01000	01001	01010	01011	01100	01101	01110	01111
Caractère	I	J	K	L	M	N	O	P
5 bits	10000	10001	10010	10011	10100	10101	10110	10111
Caractère	Q	R	S	T	U	V	W	X
5 bits	11000	11001	11010	11011	11100	11101	11110	11111
Caractère	Y	Z	.	(espace)	aucune signification (on peut, si on le souhaite, s'en servir pour encoder d'autres signes de ponctuation)			

A l'aide de la table de correspondance ci-dessus, encode le message ci-dessous en langage binaire.

Message en clair	D	I	X		M	I	N	U	T	E	S
Message codé en binaire											