

PROGRAMMATION CREATIVE AVEC SCRATCH JUNIOR

Éléments de contexte : références au programme et au socle commun

COMPÉTENCES TRAVAILLÉES	DOMAINES DU SOCLE
Pratiquer des démarches scientifiques et technologiques Proposer, avec l'aide du professeur, une démarche pour résoudre un problème ou répondre à une question de nature scientifique ou technologique : <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 	Domaine 4 : Les systèmes naturels et les systèmes techniques. <ul style="list-style-type: none"> Capacité à résoudre des problèmes
Concevoir, créer, réaliser <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 	Domaine 5 : Les représentations du monde et de l'activité humaine <ul style="list-style-type: none"> Compréhension des sociétés dans le temps et dans l'espace
Pratiquer des langages <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. 	Domaine 1 : Les langages pour penser et communiquer. <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
S'approprier des outils et des méthodes <ul style="list-style-type: none"> Organiser seul ou en groupe un espace de réalisation expérimentale. Utiliser les outils mathématiques adaptés 	Domaine 2 : Les méthodes et outils pour apprendre. <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).			

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.

Séance 1 :

Objectif : Découvrir l'interface de programmation de Scratch Junior ou de Scratch

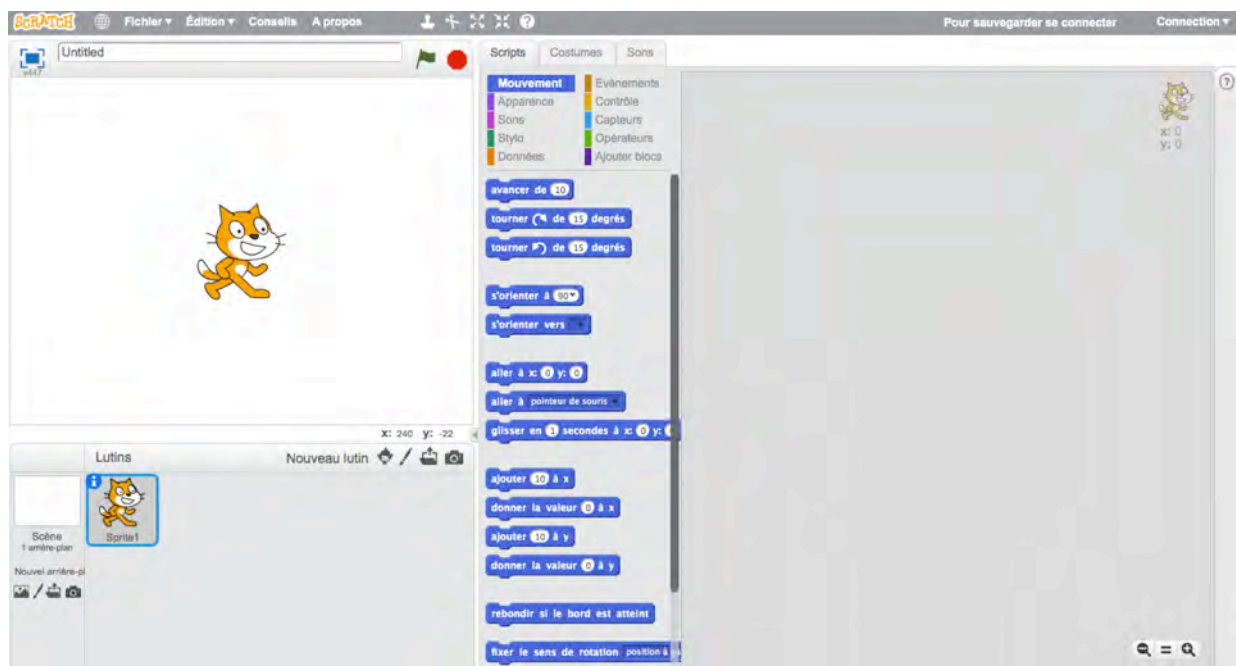
Le module est écrit pour l'utilisation de Scratch Junior. Lien pour télécharger Scratch Junior sur tablette android ou IOS

<https://www.scratchjr.org/>



Si vous utilisez Scratch, vous pouvez vous inspirer de la séquence proposée et vous aider du tutoriel proposé en annexe. Lien pour télécharger Scratch offline sur PC windows, Mac, ...

<https://scratch.mit.edu/scratch2download/>



Phase 1 : Faire découvrir un langage de programmation en plaçant directement les élèves en situation avec « scratch junior » sur les tablettes avec trois questions:

- A quoi ça sert?
- Comment ça marche?
- Questions/problèmes rencontrés?

Pendant ce temps de tâtonnement, l'enseignant essaye de repérer les questions qui se posent, les problèmes rencontrés et aider les élèves à les formuler.

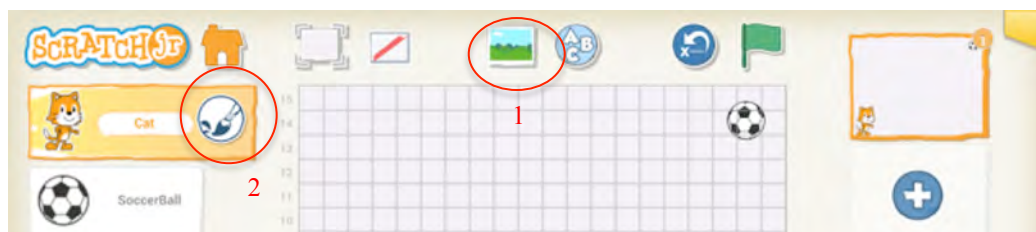
Au bout de quelques minutes on fait un premier point avec des démonstrations de groupes au TBI (quand on a un TBI et un adaptateur VGA/tablette) pour aider les groupes en panne, partager les premières découvertes.

Au cours de cette première séance de découverte, les façons d'appréhender Scratch Junior par les élèves sont variées.

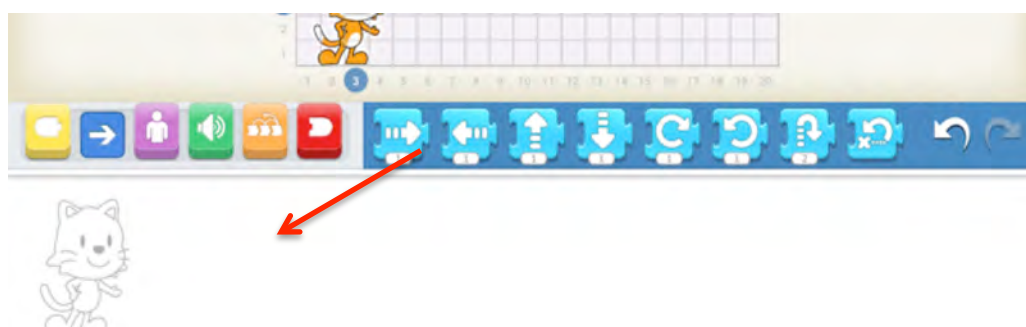
Certains vont directement aller vers le tutoriel animé présent sur la page d'accueil,



d'autres vont aller vers les fonctionnalités permettant d'insérer des paysages (1) et des personnages, vers l'assistant graphique qui permet de changer les couleurs (2), ...



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 1 : exercices)

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 : Défi

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.

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.

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).

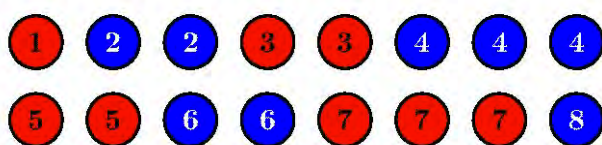
Consigne :

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.

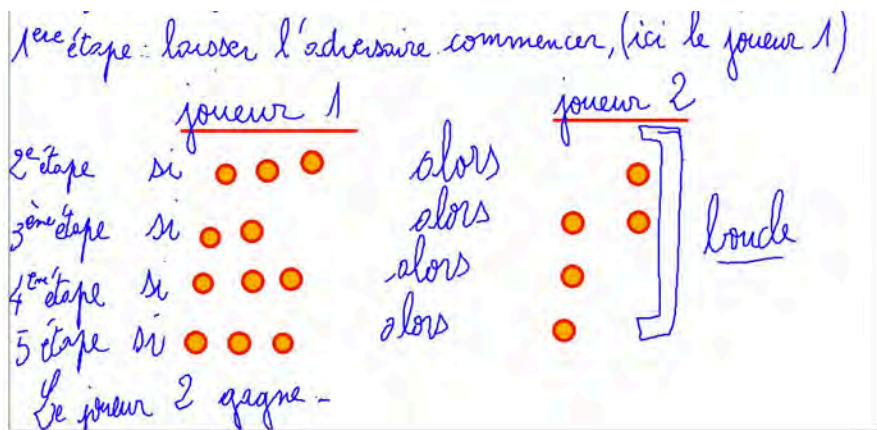
Phase 3 : Mise en situation à partir du jeu de Nim, 2^{ème} tour.

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.

Satxi 17/03/12 010
 Le jeu de Nim
 Mon programme pour gagner :
 1^{ère} étape Laisser commencer, des que
 4 jetons sur la table
 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

autres 73/16
 Le jeu de Nim
 Mon programme pas pour gagner :
 1^{ère} étape je 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

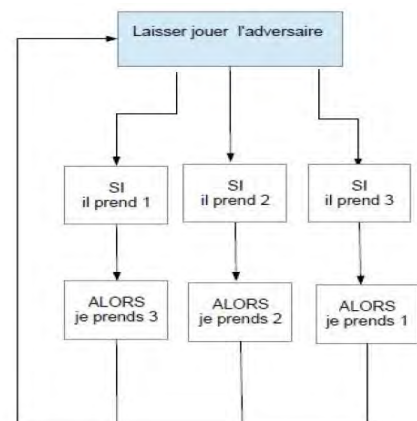
A la fin, le maître choisit un élève qui lui donne des instructions pour réaliser le programme gagnant. Le programme proposé ici comporte 5 étapes, avec répétition d'instructions conditionnelles **SI...ALORS**.



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.

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 une 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.