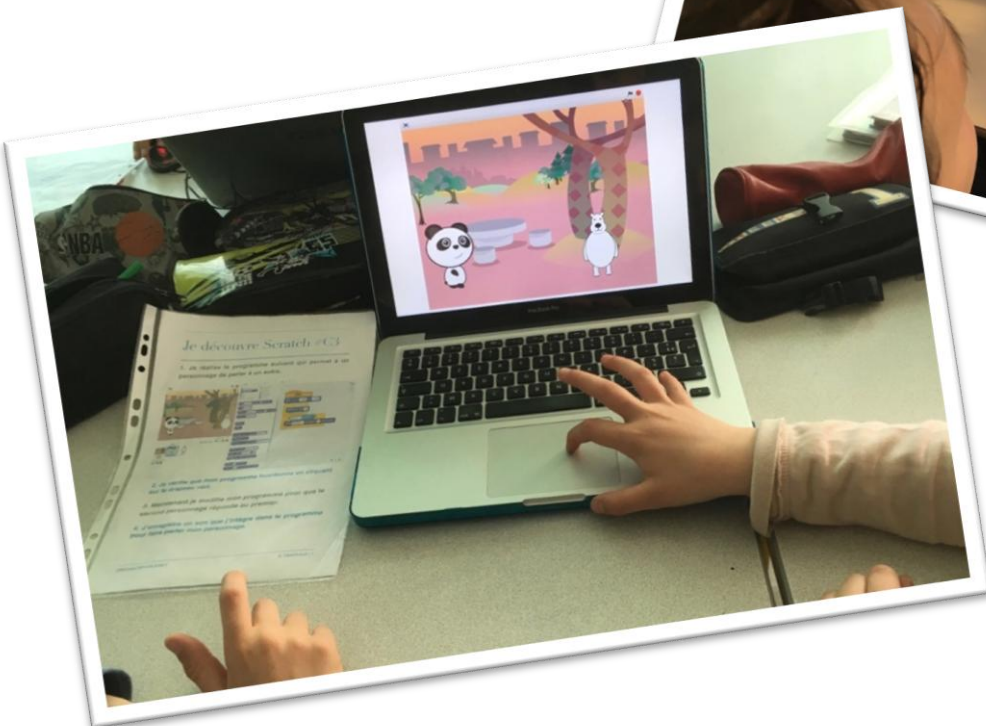


Séance de découverte et d'initiation à la programmation



Codage, programmation, algorithmie : kesako ?



Avec l'arrivée des programmes 2016, une nouvelle notion « le code informatique » vient de faire son entrée **officielle** à l'école primaire. Derrière ce terme se cachent des enjeux fondamentaux d'apprentissage, essentiels à la compréhension du monde dans lequel les élèves grandissent.

Commençons par replacer le terme de « codage » dans son contexte : le code est un aspect ponctuel de l'algorithmie qui est elle-même l'un des aspects de ce que l'on appelle la pensée informatique (concept développé par Alan Turing dans les années 50).

Pour vulgariser : la pensée informatique repose sur la notion d'algorithme, c'est à dire un enchainement mécanique d'actions ordonnées servant à résoudre un problème ou à réaliser une action globale.

La notion d'algorithme est présente partout dans notre vie quotidienne, de la recette de cuisine à la composition musicale en passant par le régulateur de vitesse de la voiture...

Elle se décompose en cinq grandes étapes :

- ↪ **Abstraction** : appréhender le problème à résoudre dans un cadre modélisant
- ↪ **Anticipation** : réfléchir aux actions à réaliser sous forme d'une série d'étapes
- ↪ **Décomposition** : scinder le problème complexe en sous problèmes simples ordonnés
- ↪ **Reconnaissance** : faire appel à ses connaissances de problème similaires
- ↪ **Généralisation** : la solution peut servir à résoudre d'autres problèmes.

La pensée informatique est donc une façon d'aborder les problèmes qui se traduit par la réalisation d'algorithme dont le code n'est que l'écriture informatique.

De la pensée informatique à l'informatique créative

Comment trouver une porte d'entrée pertinente, adaptée au milieu scolaire ?

L'élève doit pouvoir être au cœur de cet apprentissage nouveau et pour lequel ses représentations sont faibles ou inexistantes. C'est tout l'enjeu de ce que l'on appelle **l'informatique créative**.

Entrer dans une démarche d'informatique créative, c'est :

- apprendre à prévoir, anticiper des paramètres extérieurs et y faire face,
- revenir en permanence sur son travail pour voir la validation possible,
- opérer des transferts de compétences,
- construire des concepts transposables,
- comprendre comment fonctionne les choses du quotidien pour mieux comprendre le monde.

C'est donc un véritable processus créatif car les chemins pour arriver au résultat sont aussi nombreux et divers que leurs concepteurs... ☺



⇒ Cycle 2

« Mettre en œuvre un algorithme de calcul posé pour l'addition, la soustraction, la multiplication »

« Au CP, la représentation des lieux et le codage des déplacements se situent dans la classe ou dans l'école, puis dans le quartier proche, et au CE2 dans un quartier étendu ou le village. Dès le CE1, les élèves peuvent coder des déplacements à l'aide d'un logiciel de programmation adapté, ce qui les amènera au CE2 à la compréhension, et la production d'algorithmes simples. »

⇒ Cycle 3

« En CM1 et CM2 on se limitera aux signaux logiques transmettant une information qui ne peut avoir que deux valeurs, niveau haut ou niveau bas. En classe de sixième, l'algorithme en lecture introduit la notion de test d'une information (vrai ou faux) et l'exécution d'actions différentes selon le résultat du test. »

« Les élèves apprennent à connaître l'organisation d'un environnement numérique. Ils décrivent un système technique par ses composants et leurs relations. Les élèves découvrent l'algorithme en utilisant des logiciels d'applications visuelles et ludiques. Ils exploitent les moyens informatiques en pratiquant le travail collaboratif. Les élèves maîtrisent le fonctionnement de logiciels usuels et s'approprient leur fonctionnement. »

Rappel du socle commun :

« L'élève sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples. »

☞ On le voit bien, il ne s'agit aucunement de faire de nos élèves des programmeurs informaticiens mais d'offrir un cadre pédagogique réfléchi et pertinent à un apprentissage nouveau d' « initiation à la programmation ».

Ces activités nouvelles sont sources d'estime de soi, de collaboration, de confrontations, d'échanges verbaux fructueux, de travail en projet.



1) Activité débranchée : « le robot idiot »

Dispositif / Matériel nécessaire :

- 25 plots plats type coupelles [à disposer en formant un carré 5x5 à défaut 16 plots en 4x4]
- 2 ardoises par groupe de 3 ou 4 élèves
- 2 plots type cône pour marquer le départ et l'arrivée

Principe :

Dans chaque groupe, 2 ou 3 élèves jouent les « programmeurs » et programment les actions permettant un déplacement pour aller du point de Départ au point d'Arrivée en utilisant un code établi ensemble. Ce programme sera exécuté par le dernier élève du groupe qui jouera le robot.

Phase 1 : découverte

Consigne : « Vous avez devant vous le terrain de jeu d'un robot. Le robot part de ce cône (montrer le départ) et doit arriver ici (montrer le cône d'arrivée). Décrivez les actions successives que devrait réaliser ce robot pour rejoindre son plot d'arrivée en sachant que le robot se déplace uniquement de plot en plot et ne peut pas se déplacer en diagonale. »

Réponses attendues : « Il faut qu'il avance et puis après il tourne », « il avance de 4 coupelles tout droit puis il ... » etc.

→ Un élève ou le maître joue le rôle du robot afin de mettre en évidence l'importance de la précision et de la codification du langage utilisé.

Il est possible de faire une première approche où le robot agit en temps réel sous les consignes du groupe classe afin que les élèves se représentent ce qui est attendu. Ensuite, il est demandé au groupe de proposer un programme à l'oral dans son entier qui sera réalisé ensuite par le robot testeur.

Préciser que le robot ne peut avancer ET tourner en même temps. Lorsqu'il tourne, cela signifie qu'il pivot sur un quart de tour à gauche ou à droite.

Consigne : « Nous allons maintenant choisir ensemble un langage qui sera compris de tous ».

→ Se mettre d'accord sur les symboles codant les mouvements du robot (et éventuellement les actions)

Voici une proposition de langage qui peut être envisagé :



Phase 2 : appropriation du langage choisi

Les élèves se placent en groupe de 3 ou 4 élèves. Un élève jouera le rôle du robot, les autres sont les « programmeurs ». Les points D et A sont changés afin d'établir un nouveau parcours.

Consigne : « *Maintenant que nous nous sommes mis d'accord sur le langage à utiliser, proposez sur votre ardoise une programmation que votre robot va effectuer. Attention, votre programme doit être copié sur les 2 ardoises du groupe à l'identique car vous devrez remettre une de vos ardoises à votre robot afin qu'il effectue le parcours indiqué, l'autre vous servira à suivre et vérifier votre programme en temps réel. Vous avez 5 minutes de préparation.* »

Durant cette phase :

- Les élèves jouant le rôle du robot sortent ou sont isolés sans vue sur ce que préparent leurs camarades. La consigne leur est donnée de ne se déplacer que sur le code qui a été établi en commun auparavant et de refuser toute autre instruction. Ils devront suivre strictement ce qui leur a été demandé.
 - Les élèves « programmeurs » peuvent investir l'espace. Certains resteront peut-être en retrait pendant que d'autres joueront le rôle du robot pour mieux appréhender le parcours. Il est important de laisser les groupes réfléchir et proposer leur solution.

Au bout des 5 minutes, retour des robots :

Les programmeurs donnent l'ardoise contenant le programme à réaliser. Ils gardent en copie sur la seconde ardoise le même programme. Chaque robot exécute le programme imaginé par les élèves de son groupe en oralisant les commandes du déplacement écrit. Les élèves qui ont réalisé la programmation entourent fur et à mesure les actions effectuées afin de repérer celles qui posent problème.

- ☞ Ce temps doit permettre de valider ou non le programme. Les élèves auront ensuite la possibilité de modifier leur programmation et de le retenter autant de fois que nécessaire avec leur robot.

Mise en commun :

Consigne : « *Finally, que faut-il faire pour permettre à notre robot d'exécuter la tâche que nous avons imaginée ?* »

Réponses attendues : « bien réfléchir au parcours, imaginer que l'on est à la place du robot, être précis, faire par étape.... »

Institutionnalisation :

« Pour réussir ma programmation, je dois respecter le langage du programme et anticiper toutes les étapes que devra réaliser mon robot. »

Prolongements possibles :

- On place des points de passages obligés ou des bonus (balles ou autres objets) à récupérer sur le chemin.
 - Le robot suit un programme caché, les autres élèves doivent le retrouver en fonction des actions effectuées.
- ☞ Il ne faut pas hésiter à reconduire cette activité autant de fois que besoin sous sa forme initiale ou sous les prolongements proposés. Une expansion du langage peut se faire à chaque reprise de l'activité (premières boucles, « si ...alors... » Etc.)

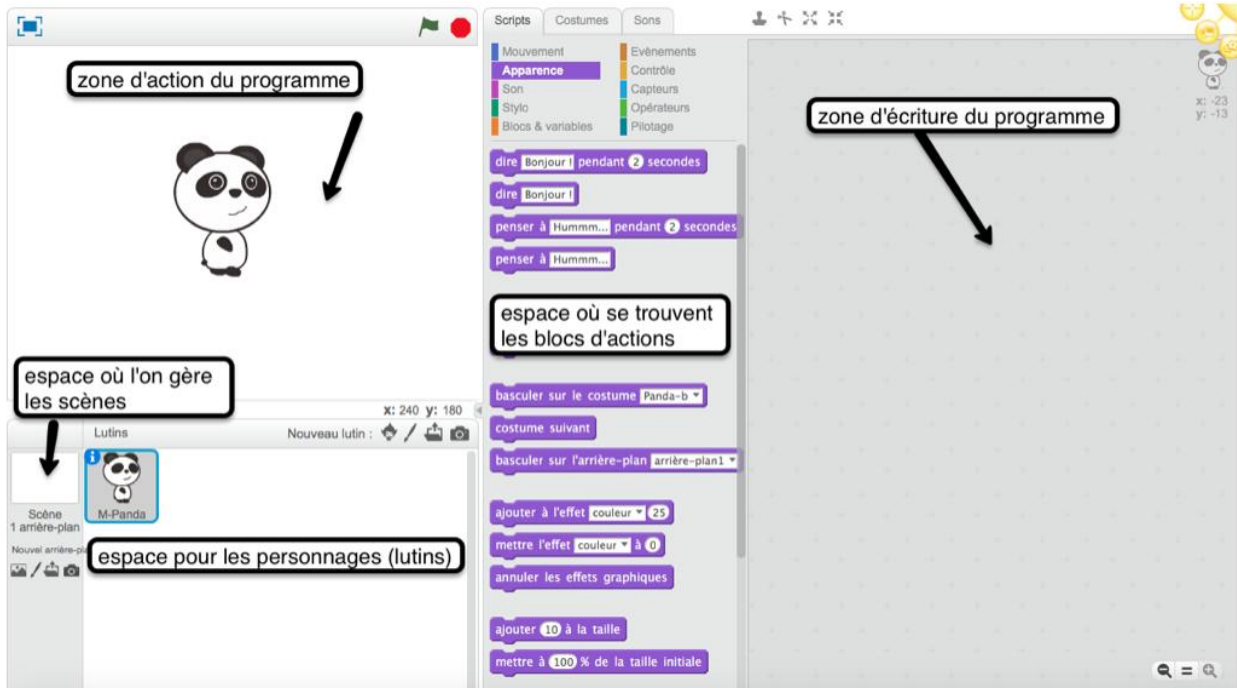


2) La découverte de l'interface « Scratch »

Matériel :

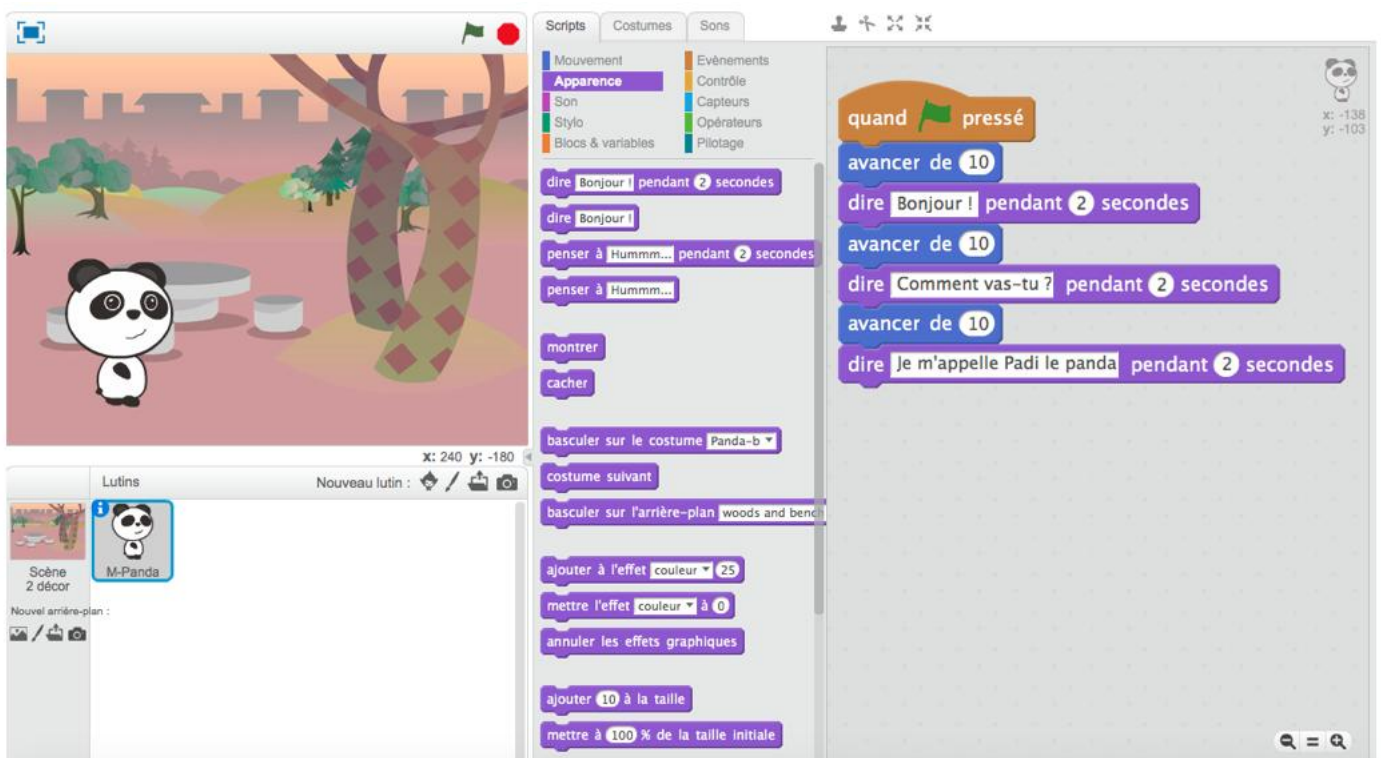
- vidéoprojecteur et recopie de l'interface vide
- 1 ordinateur pour deux élèves (afin de faciliter les échanges) avec le logiciel Scratch 2.0 ou mblocky (Scratch Jr pour tablette)

Phase 1 : Présentation rapide des différentes zones : changer les lutins, le décor, les blocs de programmation. (5')



Phase 2 : Vidéoprojection le programme à recopier.

▪ Proposition programme niveau 1 #C2 (support : ordinateur)



▪ Proposition programme niveau 2 #C3 (support : ordinateur)



▪ Proposition programme niveau 1 #C1 - (support : Tablette)



Bonjour !
Je m'appelle Scratch.
Allons dans la ville !



Consigne : « Selon vous, que demande-t-on au panda (ou au chat) de faire dans ce programme? »

Réponses attendues : « Le panda va dire bonjour, il va avancer... »

☞ Le rôle de l'enseignant est ici de faire préciser et expliciter aux élèves leurs propos.

« Comment le sais-tu ? Quels sont les éléments qui te permettent de dire cela ?... »

Il est important que les élèves puissent faire du lien entre les blocs présentés et les actions qui sont attendues. En anticipant ce qui est demandé, ils construisent le sens et pourront reproduire plus efficacement le modèle présenté.

Puis, une fois cette étape préalable réalisée, il faut mettre très rapidement les élèves à l'activité. En binôme, ils sont amenés à réaliser leur premier programme (cf fiche #C1, #C2 ou #C3)

Phase 3 : Temps de réalisation (20 à 25')

Les élèves doivent effectuer plusieurs tâches :

- choisir les lutins
- choisir l'arrière-plan
- recopier à l'identique le programme
- modifier des contenus de bloc
- vérifier son exécution

Il leur est proposé aussi de modifier les programmes par amplification et modification du script afin de le personnaliser.

Le rôle de l'enseignant sera ici d'accompagner en répondant aux sollicitations des élèves, non pas pour donner la solution mais pour les guider vers elle par le questionnement. Il ne faut pas hésiter à laisser les enfants réfléchir, tâtonner dans leur recherche... À travers cette activité, ils vont découvrir les différentes zones, les manipulations de blocs, les liens entre les types de blocs (mouvement,...) et les blocs qui les constituent, les emboitements, les modifications,

Prolongements possibles:

Ce programme pourra servir de base à un travail d'écriture narrative en français ou à un travail de rebrassage en langues vivantes.



Quelques ressources pour aller plus loin:



Présentation du jeu du robot idiot : <https://pixees.fr/dis-maman-ou-papa-cest-quoi-un-algorithme-dans-ce-monde-numerique-%E2%80%A8>

Pour aller plus loin avec Scratch :



Guide de l'informatique créative :

https://iww.inria.fr/mecsci/CreativeComputingFr/CreativeComputing20140806_FR_tfrancais.pdf

Cartes missions scratch (chez CANOPE 78)

Pour ScratchJr : <http://www.mde78.ac-versailles.fr/spip.php?article1161>

Pour Scratch : <http://www.mde78.ac-versailles.fr/spip.php?article1158>

Quelques applications pour tablettes



Tickle



Scratch Jr



Lightbot



Hopscotch



Tynker



Run Marco

Quelques exemples de productions de classe sous scratch :

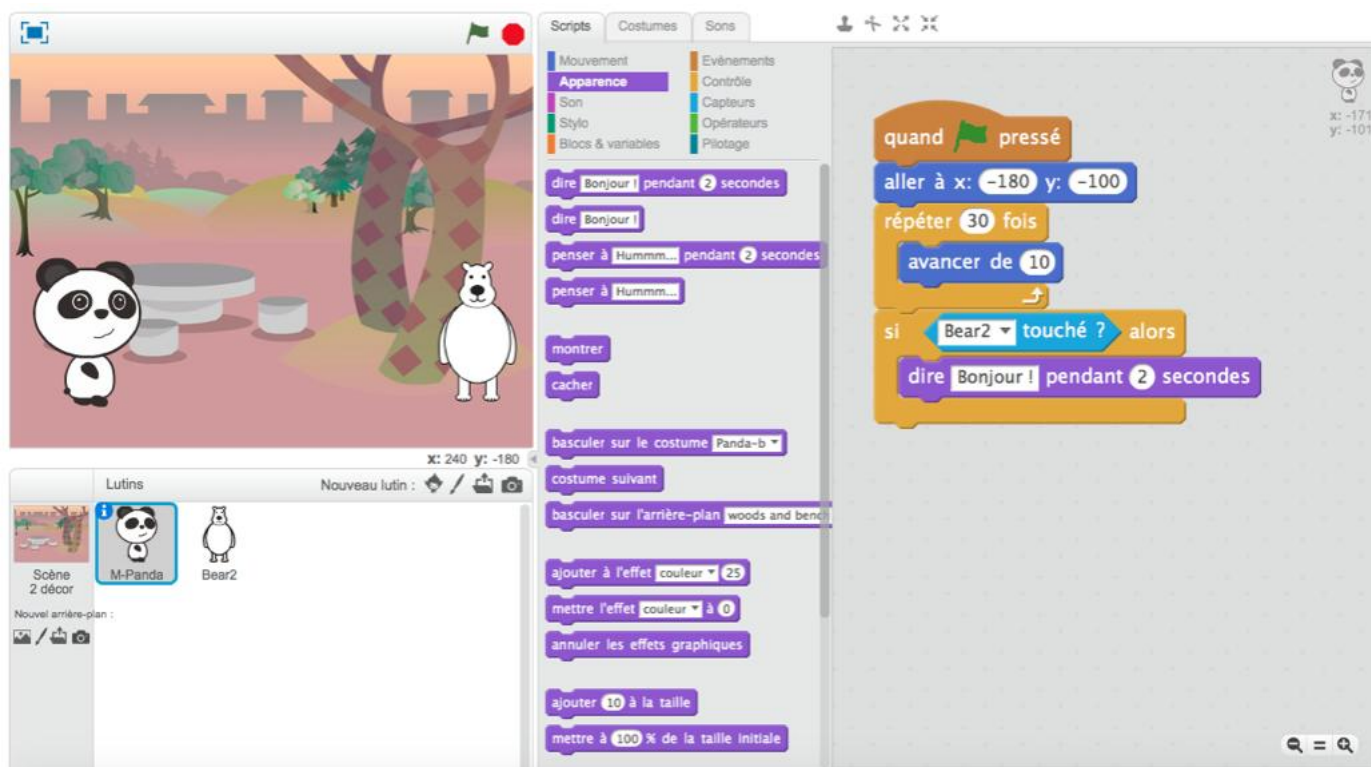
- Rituels en anglais : <https://scratch.mit.edu/projects/91496272/>
- Rituels en allemand : <https://scratch.mit.edu/projects/100370458/>
- Test APER : <https://scratch.mit.edu/projects/103843931/>
- Jeu du caissier (résolution de problème) : <https://scratch.mit.edu/projects/102247117/>



JE DÉCOUVRE SCRATCH #C3



1. Je réalise le programme suivant qui permet à un personnage de parler à un autre.



2. Je vérifie que mon programme fonctionne en cliquant sur le drapeau vert.

3. Maintenant je modifie mon programme pour que le second personnage réponde au premier.

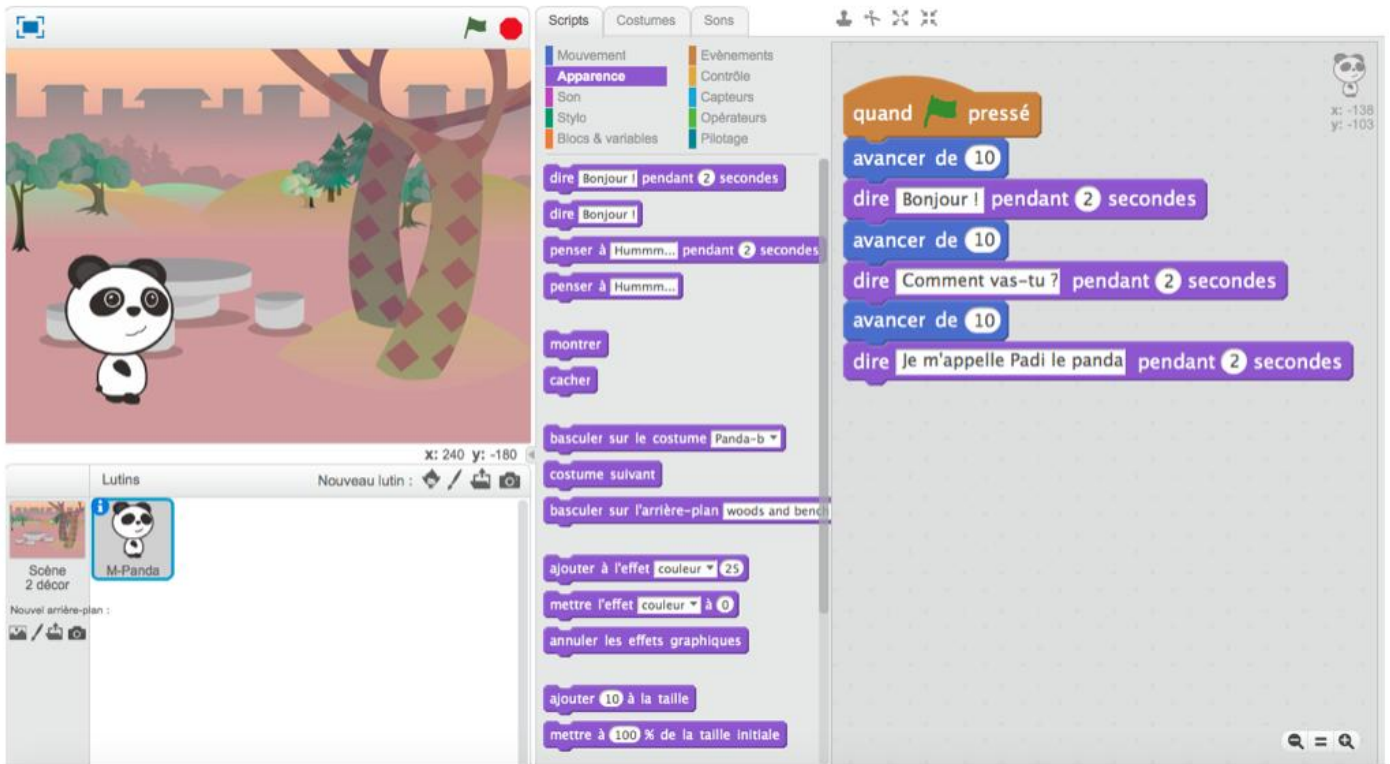
4. J'enregistre un son que j'intègre dans le programme pour faire parler mon personnage.



JE DÉCOUVRE SCRATCH #C2



1. Je réalise le programme suivant qui permet à un personnage de parler et d'avancer.



2. Je vérifie que mon programme fonctionne en cliquant sur le drapeau vert.
3. Maintenant je modifie mon programme pour continuer à lui faire dire de nouvelles phrases.
4. J'enregistre un son que j'intègre dans le programme pour faire parler mon personnage.



JE DÉCOUVRE SCRATCHJR #C1

1. Je réalise le programme suivant qui permet à un personnage de se parler, de se déplacer pour se rendre dans un autre décor.



Bonjour !
Je m'appelle Scratch.
Allons dans la ville !

2. Je vérifie que mon programme fonctionne en cliquant sur le drapeau vert.
3. J'enregistre un son que j'intègre dans le programme pour faire parler mon personnage.
4. Maintenant je prolonge mon programme pour continuer les aventures dans le nouveau décor.

