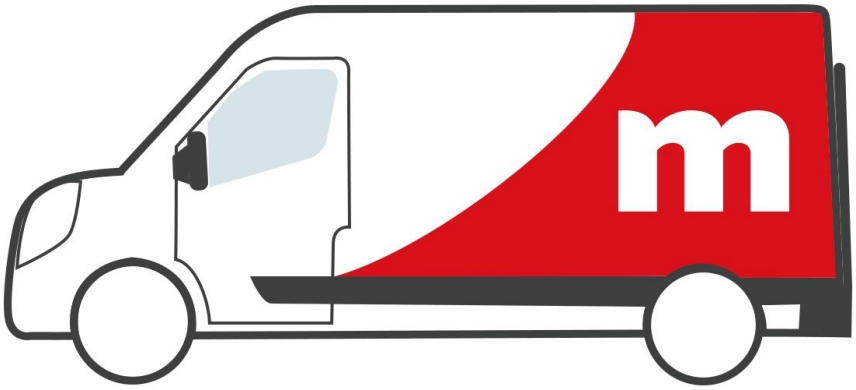
LA CASEMATE NOMADE



Apprendre à coder

Scénario 2 : crée ton projet

Par Catherine Demarcq et Tristan Hamel, La Casemate

8 - 14 ans | 2h | 12 participants | Programmation

Modalités

Mots clés :
Programmation, informatique créative

Niveau scolaire :
Primaire, Collège, Informel

Encadrement :
Autonomie

Coût :
0€

Groupe :
Individuel/Groupe

Lieu d'animation :
Petite salle à l'intérieur avec une prise électrique ou Salle informatique

Compétences :
Utiliser des pensées mathématiques ou informatiques

Prérequis de l'animateur :
Connaissance de Scratch

Matériel nécessaire

- Fiches d'activités plastifiées :
    - https://resources.scratch.mit.edu/www/cards/fr/Scratch2Cards.pdf
    - https://resources.scratch.mit.edu/www/guides/fr/Getting-Started-Guide-Scratch2.pdf
- 6 ordinateurs avec internet ou scratch (https://scratch.mit.edu/download) pour un groupe de 12 enfants, les enfants travaillent en binôme , avec souris et tapis de souris obligatoires
- Papier vierge, feutres
- Prise multiple et rallonge

Cette activité fait partie d'une série de 3 scénarios "Apprendre à coder" qui permettent de découvrir Scratch.

Objectifs

- Procurer une première approche de l'algorithmique à des jeunes
- Apporter les bases de la programmation
- Favoriser le développement d'une pensée créative
- Découvrir l'intérêt de partager avec une communauté

Objectifs d'apprentissage

- Faire comprendre la notion d'algorithmes et les notions sous jacentes : action, séquence, boucle, structure conditionnelle
- Comprendre l'algorithme et pourquoi il marche / ou pas
- Construire un algorithme pour réaliser un projet
- Appliquer ce que l'on a appris pour pouvoir programmer un robot (thymio par exemple)
- Développer sa créativité

Evaluation

- Je peux expliquer ce que j'ai fait en utilisant des verbes actions : avancer, tourner ..
- Je sais choisir dans le menu des instructions en fonction de ce que je veux faire

Informations préalables

Scratch est un logiciel libre conçu pour initier les enfants dès l'âge de 8 ans à des concepts fondamentaux en mathématiques et en informatique. Il repose sur une approche ludique de l'algorithmique, pour les aider à créer, à raisonner et à coopérer.Il favorise également leur partage sur le Web. A partir de 2007, le site Web a été ouvert afin de permettre à tous d'une part, de publier, donc de faire partager, ses projets sur le Web et d'autre part d'apporter une aide à la mise en œuvre de Scratch (source : scratchfr.free.fr).

Des tutoriels en ligne existent pour apprendre à maîtriser Scratch. Des tutoriels simples sont accessible par exemple sur magicmakers ou scratch.mit.edu.

L'animateur dispose du guide de démarrage :
https://resources.scratch.mit.edu/www/guides/fr/Getting-Started-Guide-Scratch2.pdf

Une formation d'une demi-journée sur le logiciel permet une prise en main optimale.

Ce scénario est adapté pour l'apprentissage de Scratch. Si les enfants connaissent déjà l'interface, passer directement au scénario 2.

Consulter les exemples de projets Scratch pour inspiration
⇒ https://scratch.mit.edu/starter_projects/

Description complète de l'activité

Scénario 2 : Développer sa créativité à l'aide de Scratch : Crée ton projet. (2h)

Public : Atelier proposé à des enfants qui ont peu ou pas pratiqué scratch.

Objectif : Laisser aux enfants le temps de créer un projet n'utilisant que les 10 blocs Scratch suivants : aller à, glisser, dire, montrer, cacher, mettre à <n> % de la taille, jouer le son jusqu'au bout, quand ce lutin est cliqué, attendre et répéter.

Les encourager à faire des tests et à résoudre les problèmes rencontrés en "compartimentant" ces derniers.

Etape 1.Présentation de l'outil scratch et de ses possibilités (10 à 15 ')

Organisation : groupe de 12 enfants (2 par poste)

Déroulement :

- L'animateur fait installer les enfants 2 par 2 devant les ordinateurs. Il leur fait regarder la vidéo du site : https://scratch.mit.edu/about/
- L'animateur présente l'interface et les principales fonctionnalités
- Les participants créé un compte Scratch, d'un projet (nécessite un accès à INTERNET!) ⇒ Cette étape n'est pas obligatoire
- Les participants créent un lutin
- Les participants créent un arrière plan
- Utilisation des blocs (insertion, modification, exécution, suppression)
- Test/débuggage

Etape 2. Défi : "Fais bouger le lutin"

Organisation : groupes hétérogènes de 2 enfants (1 qui donne les instructions, 1 qui exécute avec inversion des rôles)

Déroulement :

- Création de scénarios (2x10')
    - Celui qui donne les instructions crée un scénario et le dicte à l'exécutant qui le programme
    - Inversion des rôles
- Echanges (10')

L'animateur pose des questions aux groupes, sur les réussites, les difficultés rencontrées, les questionnements…

- Techniques de résolution de problème: (10')

L'animateur propose 8 règles de résolution de problème qui permettront de résoudre le problème tout en réussissant et en progressant. Elles permettront aussi de renforcer la confiance en eux des participants.

Règle 1 : Avoir un plan

Ton plan peut être modifié à un moment donné, ou tu peux abandonner ton plan original et en imaginer un autre.

Planifier te permet également de te fixer des objectifs intermédiaires et de les atteindre. Sans plan, tu as un seul objectif : résoudre tout le problème.

Lorsque tu auras atteint ton objectif intermédiaire, tu seras en mesure de vérifier les éléments de ton plan, de réussir et ainsi de gagner en confiance.

Règle 2 : Reformuler le problème

Reformuler le problème te montre parfois que le but n'est pas ce que tu pensais qu'il était.

Ceci peut être ta première étape et être considéré comme un progrès.

De plus cela peut te permettre de diviser ou réduire le problème.

Règle 3 : Décomposer le problème

Trouver un moyen de décomposer le problème en problèmes simples peut faciliter le problème.

Si tu peux diviser un problème en deux morceaux, tu pourrais penser que chaque morceau serait moitié aussi difficile à résoudre que le problème de départ, mais en général, c'est encore plus facile que cela.

Règle 4 : Commencer par ce que tu sais

Lors de la programmation, tu devrais essayer de commencer avec ce que tu sais déjà faire et travailler à partir de là. Ainsi, tu construis la confiance et l'élan vers l'objectif.

Règle 5 : Simplifier le problème

Face à un problème que tu ne parviens pas à résoudre, tu réduis la portée du problème en ajoutant ou en supprimant des contraintes pour produire un problème que tu sais résoudre.

Cela te permet de travailler sur un problème plus simple, même si tu ne peux trouver un moyen de diviser le problème en étapes.

Simplifier le problème te permet aussi de déterminer exactement où se trouve la difficulté.

Règle 6 : Rechercher les analogies

Une analogie, est une similitude entre un problème courant et un problème déjà résolu qui peut t'aider à résoudre le problème actuel.

Il est nécessaire d'avoir un catalogue de solutions de référence que tu vas enrichir au fur et à mesure.

Règle 7 : Expérimenter

Parfois, la meilleure façon de faire des progrès est d'essayer des choses et d'observer les résultats. Note qu'expérimenter n'est pas la même chose que deviner.

Tu fais l'hypothèse de ce qui se passera quand un certain code sera exécuté, essaie-le et vérifie si ton hypothèse est correcte. À partir de ces observations, tu obtiendras des informations qui t'aideront à résoudre le problème initial.

Règle 8 : Ne pas se frustrer

Lorsque tu es frustré, tu ne pensez pas aussi clairement, tu ne travaillerez pas aussi efficacement, et tout prendra plus de temps et semblera plus difficile. Pire encore, la frustration peut se terminer en colère.

En fin de compte, éviter la frustration est une décision que vous dois prendre.

Tout d'abord, aie un plan et suis-le,alors tu feras des progrès. Si tu as exécuté toutes les étapes de ton plan initial et que tu n'es toujours pas prêt à commencer de programmer, alors il est temps de faire un autre plan.

En outre, quand il s'agit de se sentir frustré ou de prendre une pause, prends une pause et ne pense pas au problème jusqu'à ce que votre pause soit terminée.

Toutes ces règles te permettront de résoudre le problème tout en réussissant, en progressant. Elles te permettront aussi de renforcer ta confiance en toi.

Conclusion

L'activité permet aux participants de découvrir scratch et de développer des connaissances et des compétences en codage.

Informations complémentaires

Supports :

- Vidéo de présentation de Scratch
- Fiches activité
- Exemples de projets Scratch ⇒ https://scratch.mit.edu/starter_projects/

Connections avec les programmes scolaires

Dans le programme du cycle 4 il est spécifié qu' "Un enseignement d'informatique est dispensé à la fois dans le cadre des mathématiques et de la technologie."

Cet enseignement "permet d'acquérir des méthodes qui construisent la pensée algorithmique et développe des compétences dans la représentation de l'information et de son traitement, la résolution de problèmes, le contrôle des résultats."

Le programme de technologie du cycle 4, dans la partie "L'informatique et la programmation" il y a dans les attendus de fin de cycle "Écrire, mettre au point et exécuter un programme".

- Notion d'algorithme et de programme
- Notion de variable informatique
- Déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles.

Repères de progressivité

Repères de progressivité :
En 5ème : traitement, mise au point et exécution de programme simple avec un nombre limité de variables d'entrée et de sortie, développement de programmes avec des boucles itératives.
En 4ème : traitement, mise au point et exécution de programme avec introduction de plusieurs variables d'entrée et de sortie
En 3ème : introduction du comptage et de plusieurs boucles conditionnels imbriqués, décomposition en plusieurs sous-problèmes

Pour aller plus loin

Livres :

Scratch pour les kids, The Lead Project, Eyrolles (dès 8 ans)

J'apprends à programmer avec Scratch, Rosie Dickins, Jonathan Melmoth, Louie Stowell, Shaw Nielsen, 2016, Usborne

Apprends à programmer avec Scratch : Crée tes jeux et tes animations !, Liliane Khamsay, Claude Terosier, Gilles Capelle, 2016, Gallimard Jeunesse

Informatique créative, Karen Brennan, Christan Balch, Michelle Chung, 2015, Harvard Graduate School of Education https://alain-michel.canoprof.fr/eleve/ateliers-numeriques/code-programmation/atelier_code_programmation/res/CreativeComputing20140806_FR.pdf

Références

Sites avec des activités scratch (en anglais) :
http://scratched.gse.harvard.edu/resources/
https://scratch.mit.edu/

Remerciements :
Ophélie Bertossi, stagiaire en médiation culturelle et scientifique à La Casemate, pour la relecture et la mise en page de l'activité.

Izimakers : Source et inspiration
http://izimakers.com/blog/8-regles-pour-penser-comme-un-programmeur/

Cette activité a été réalisée dans le cadre du projet La Casemate Nomade, porté par La Casemate.