

BANDE DESSINÉE POUR LA VULGARISATION
SCIENTIFIQUE DE L'INFORMATIQUE

SÉRIE **7**

Pensée
informatique



Volume 6

**DÉFIS DE PROGRAMMATION CRÉATIVE ET
PENSÉE INFORMATIQUE
DU CONTE AU CODE AVEC SCRATCH ET VIBOT**



Margarida Romero
Maria Augusta Silveira Netto Nunes
José Humberto dos Santos Junior
Luís Antônio dos Santos Silva
Aurélie Roy
Alexandre Lepage

RECTEUR

Prof. Dr. Angelo Roberto Antonioli

VICE-RECTEUR

Prof. Dra. Iara Campelo

COUVERTURE, ADAPTATION DE TRAVAUX DU BD ORIGINAUX (ILLUSTRATIONS DE CARACTÈRES, RÉDACTION ÉLECTRONIQUE ET ADAPTATION POUR LE BD)

José Humberto dos Santos Júnior

RESPONSABLE DE LA DE L'ADAPTATION CONCEPTUELLE

Maria Augusta Silveira Netto Nunes

COLLABORATEUR: ADAPTATION DES ÉTUDES, PROJETS DE SCRATCH ET FORMULAIRES D'ÉVALUATION #5c21(BRASIL)

Luís Antonio Santos Silva

CONCEPTION ORIGINALE

Cette bande dessinée est une reproduction adaptée, transformée en format bande dessinée et traduite de la version 2016 de l'œuvre "Défis de création créative: du contenu au code avec Scratch et Vibot". Disponible à: <https://goo.gl/bBWGIk> par Romero, M., Roy, A, Lepage, A. développé par l'Université Laval à Québec.

Os personagens e as situações desta obra são reais apenas no universo da ficção; não se referem a pessoas e fatos concretos, e não emitem opinião sobre eles.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE

D313d Défis de programmation créative et pensée informatique : du conte au code avec scratch et vibot [recurso eletrônico] / Margarida Romero ... [et al.]. – Porto Alegre : SBC, 2020.
72 p. : il. – (Bande dessinée pourp la vulgarisation Scientifique de L'informatique). Série 7, Pensée informatique ; v. 6)
Tradução de: Desafios de programação criativa e pensamento computacional : do conto ao código com scratch e vibot.
ISBN 978-85-7669-492-2

1. Linguagem de programação (Computadores). 2. Scratch (Linguagem de programação de computadores). 3. Computação. I. Romero, Margarida. II. Série. III. Universidade Federal de Sergipe.
CDU 004.438(059)



Margarida Romero
Maria Augusta Silveira Netto Nunes
José Humberto dos Santos Junior
Luís Antônio dos Santos Silva
Aurélie Roy
Alexandre Lepage

BANDE DESSINÉE POUR LA VULGARISATION SCIENTIFIQUE DE L'INFORMATIQUE

Série 7: Pensée informatique

**Volume 6: DÉFIS DE PROGRAMMATION CRÉATIVE ET PENSÉE INFORMATIQUE:
DU CONTE AU CODE AVEC SCRATCH ET VIBOT**

Porto Alegre/RS
Sociedade Brasileira de Computação
2020

Présentation

Cette bande dessinée a été rendue possible grâce au projet de Bourse de Productivité CNPq - DTII n° 306576 / 2016-3, coordonné par Prof^a. Maria Augusta S. N. Nunes en développement au Département d'informatique (DCOMP) / "Programa de Pós-graduação em Ciência da Computação" (PROCC) - UFS. Les publics cibles des bandes dessinées sont les jeunes et les enfants. Leur objectif général est de susciter l'intérêt pour le domaine de l'informatique. Cette bande dessinée, volume 6 de la série 7, est une reproduction adaptée du guide de défis "Défis de programmation créative: du conte au code avec Scratch et Vibot", produit par Margarida Romero et ses collaborateurs en 2016 à l'Université Laval à Québec (Romero et al., 2016). Dans cette sixième bande dessinée de la série Almanach pour la Popularisation des Sciences Informatiques, nous présentons plusieurs défis en matière de programmation créative et de développement de la pensée informatique dans les loisirs pouvant être accomplis à l'aide d'un logiciel de programmation visuelle Scratch plug-in ou en développant des alternatives créatives de manière créative et débranché. Dans ce guide, qui semble nous aider, est le chef de la Ligue des robots, responsable du développement de la pensée informatique dans le monde, Vibot, le représentant canadien / français né en 2016.

(les Auteurs)

(Les informations contenues dans ce document sont la responsabilité de leurs auteurs.)

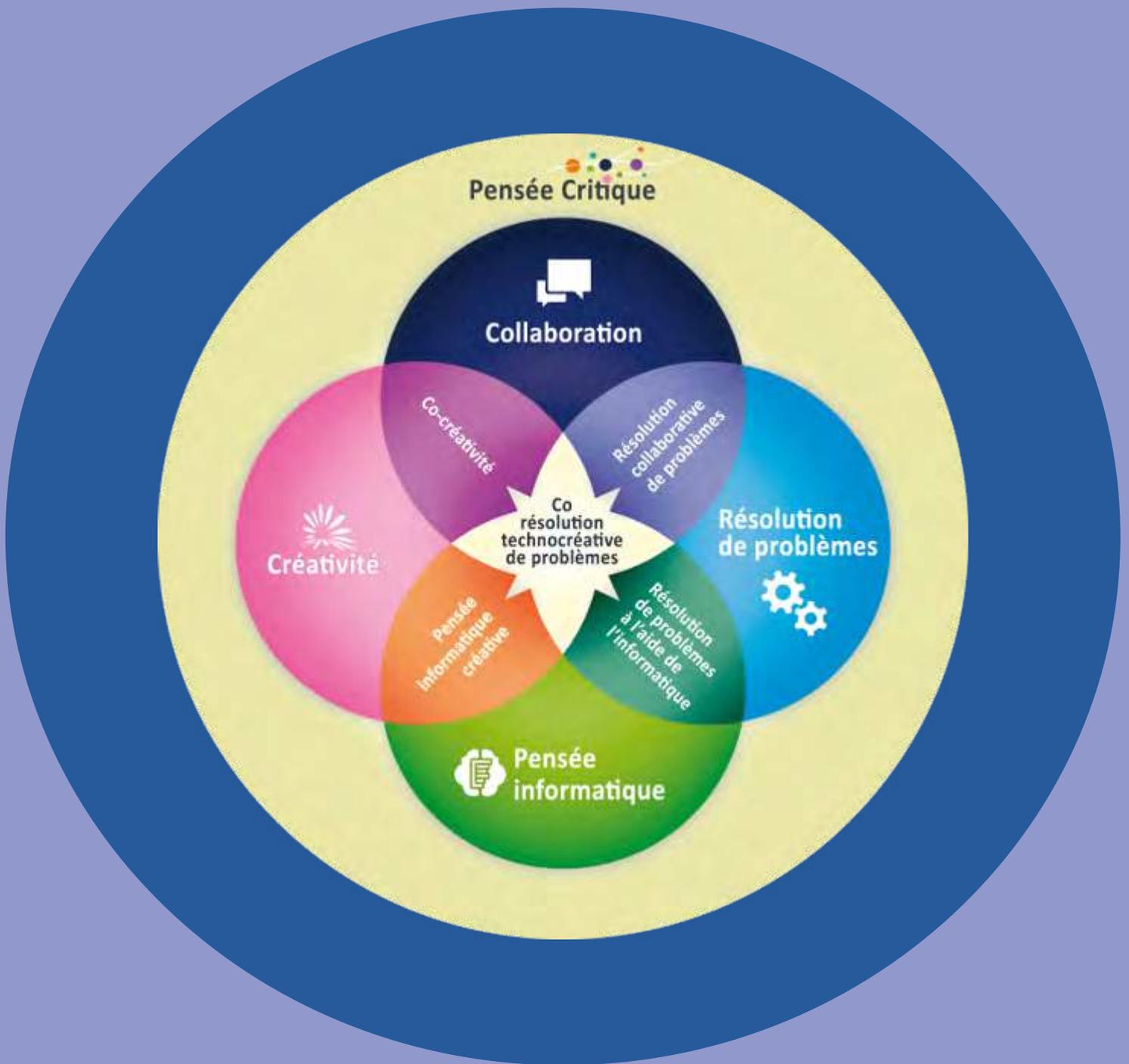
DÉFIS DE PROGRAMMATION CRÉATIVE ET PENSÉE INFORMATIQUE:

Du conte au code avec Scratch et Vibot.

Cette sixième bande dessinée de la Série "Bande dessinée pour la Vulgarisation Scientifique de l'Informatique" présente une série de défis de programmation créative et développement du pensée informatique qui peuvent être réalisés avec le logiciel de programmation visuelle Scratch. Avant de présenter les défis nous introduisons la programmation créative, le pensée informatique et les ressources qui peuvent permettre de vous familiariser avec Scratch.

INTRODUCTION À LA PROGRAMMATION CRÉATIVE

La **programmation créative** vise engager les participants, tout d'abord, dans une démarche critique, empathique et créative de **résolution de problèmes d'une certaine complexité et authenticité**, pour ensuite les responsabiliser de la **création d'une solution techno-créative**, qui peut faire appel à l'**usage des stratégies et des processus des sciences informatiques** pour la création d'une ou plusieurs solutions. La programmation créative engage la **pensée informatique**, un ensemble de stratégies cognitives et métacognitives pour la **résolution créative de problèmes** qui fait appel aux **stratégies et aux processus des sciences informatiques** (Romero, 2016).



Dans le cadre des compétences clés pour le 21e siècle #5c21, comme suit: (1) pensée critique (PC); (2) collaboration (CO); (3) créativité (Crea), (4) résolution de problèmes (RP) et (5) **pensée informatique** (PI ou CT(Computational Thinking)). Dans l'image ci-dessus, la pensée informatique est en lien avec la **créativité** et avec la **résolution de problèmes**.

#VibotORobo



Avant de réaliser les défis de programmation créative et pensée informatique nous vous invitons à lire le conte original d'introduction à la programmation Vibot le robot (disponible en format numérique ou papier); ou dans la version

adaptée, Betabot - le représentant brésilien de la League of Bots pour le développement de la pensée informatique au Brésil (en portugais disponible en numérique).

Vibot le robot (Romero, 2016, illustré par Loufane, Publications du Québec) est un livre pour les enfants de 7 à 77 ans qui permet de faire une introduction à la programmation et à la robotique. Les héros du livre reçoivent un robot en cadeau. Ils devront apprendre à le programmer pour jouer avec lui. Heureusement, ils peuvent compter sur l'aide de mamie Ada, leur e-mamie-mathématicienne. L'histoire présente des textes en français et leur correspondance dans des blocs de programmation Scratch.

Nous, les Brésiliens, avons créé le nouveau personnage, le robot Betabot, qui est le représentant brésilien de la Ligue. Vibot est le représentant de la Ligue canadienne / française, il est le premier personnage de la Ligue de Bot, il est né en 2016, il est considéré comme le chef de la Ligue. La Ligue est chargée de développer la pensée informatique dans le monde. Betabot, née en 2019, est la représentante brésilienne de la Ligue. Elle est une entité virtuelle pouvant accompagner les enfants sur diverses plates-formes, que ce soit dans des robots physiques, des téléphones portables, des technologies / montres intelligentes, des vêtements, des tablettes, des ordinateurs, etc. Il agit conformément à sa mission dans la Ligue de pensée informatique (L.P.I.) et repose également sur son apprentissage constant en aidant les apprenants et les autres utilisateurs à travers diverses interfaces. Agissant dans la région définie pour elle au sein de la Ligue (L.P.I.).

* Vous pouvez également lire les bandes dessinées Série 7, les numéros 1, 2, 3 et 4 qui parlent de la pensée informatique et de Scratch, ainsi que les bandes dessinées Série 1 qui parlent un peu du langage Scratch.

BETABOT



Betabot - le représentant de la Ligue brésilienne des Bots pour le développement de la pensée informatique au Brésil est une bande dessinée destinée aux enfants âgés de 7 à 77 ans qui introduit la programmation et la robotique.

Les personnages de bandes dessinées reçoivent un robot en cadeau. Ils devront apprendre à le programmer pour pouvoir jouer avec. Heureusement, ils peuvent compter sur leur aide, et Rafaela, experte en programmation et mathématique, dispose toujours d'une chaîne YouTube appelée "Fala aí Geek", qui fournit des conseils sur la pensée informatique, la programmation et les mathématiques. (*Série 7, Volume 1 des l'Almanach*). L'histoire de Betabot présente des textes en portugais et son correspondant dans des blocs de programmation Scratch.

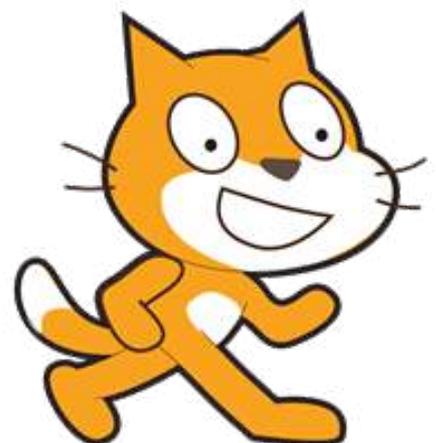


LE LOGICIEL DE PROGRAMMATION VISUELLE SCRATCH

Scratch est un logiciel libre conçu pour développer des projets créatifs avec des blocs de programmation. Les blocs peuvent s'organiser comme un casse-tête dans l'espace de programmation et les lutins et fonds d'écran peuvent être modifiés pour développer des animations et même des mini-jeux comme Mario Bros.

Le site Web <http://scratch.mit.edu> permet de consulter des milliers de projets existants et même de les remixer. Il permet également de créer des comptes de type enseignant ou de type participant afin de publier et partager des projets sur le Web.

Pour vous familiariser avec Scratch et la programmation créative nous vous invitons à lire le guide sur [la programmation créative avec Scratch de Brennan, Balch et Chung \(2014\)](#). Vous trouverez également des tutoriels vidéos sur Scratch à l'adresse suivante: [Vidéo-tutoriels Scratch](#).



STUDIO VIBOT, DES RESSOURCES GRATUITES EN LIGNE

Le studio Vibot présente des ressources en lien aux défis de programmation créative Scratch. Il est disponible à l'adresse :

En français: <https://scratch.mit.edu/studios/1999251/>

En portugais: <https://scratch.mit.edu/studios/8379890/>



The screenshot shows the Scratch Studio Vibot interface. At the top, there is a navigation bar with the Scratch logo and links for 'Créer', 'Explorer', 'Discuter', 'À propos', 'Aide', 'Recherche', 'Rejoindre Scratch', and 'Se connecter'. The main heading is '#Vibot, the robot / #Vibot, le robot' with '(15 Followers)' to the right. Below the heading are tabs for 'Projets (31)', 'Commentaires (1)', 'Curateurs', and 'Activité'. The main content area displays a grid of project thumbnails. On the left, there is a sidebar with a book cover for 'Vibot, le robot' by Romero (2016), illustrated by Loufane. The book cover shows a blue robot with orange antennae. The text in the sidebar reads: 'Mis à jour 30 Mai 2016', 'Vibot, le robot (Romero, 2016, illustré par Loufane, Publications du Québec) est un livre pour les enfants de 7 à 77 ans qui permet de faire une introduction à la programmation et à la robotique. Les héros du livre reçoivent un robot en...'. The project grid contains eight thumbnails, each with a title and author. The titles are: 'CC05.en Creative chal...', 'CC04.en. Creative cha...', 'CC03.en,Creative chal...', 'CC02.en. Creative cha...', 'CC01.en. Creative cha...', 'DC01.fr. Défi créatif #1...', 'DC02.fr. Défi créatif: D...', and 'DC04.fr. Défi créatif: D...'. The authors listed are 'par ejonnie' and 'par aurelie101'.

INTRODUCTION AUX DÉFIS DE PROGRAMMATION CRÉATIVE

Les défis de programmation créative présentent une petite histoire mettant en scène le chat **Scratch** et **Vibot le robot** (Betabot n'apparaît toujours pas car il vient de naître, mais nous aurons bientôt une série de défis à relever avec le représentant brésilien de la Ligue de Bot - Ligue de la Pensée Informatique (L.P.I.)). Pour chaque défi, nous invitons à analyser l'histoire, à identifier les composantes (personnages, décor, actions, ...) et ensuite à les programmer sur Scratch en remixant le canevas qui est proposé pour chaque défi créatif.

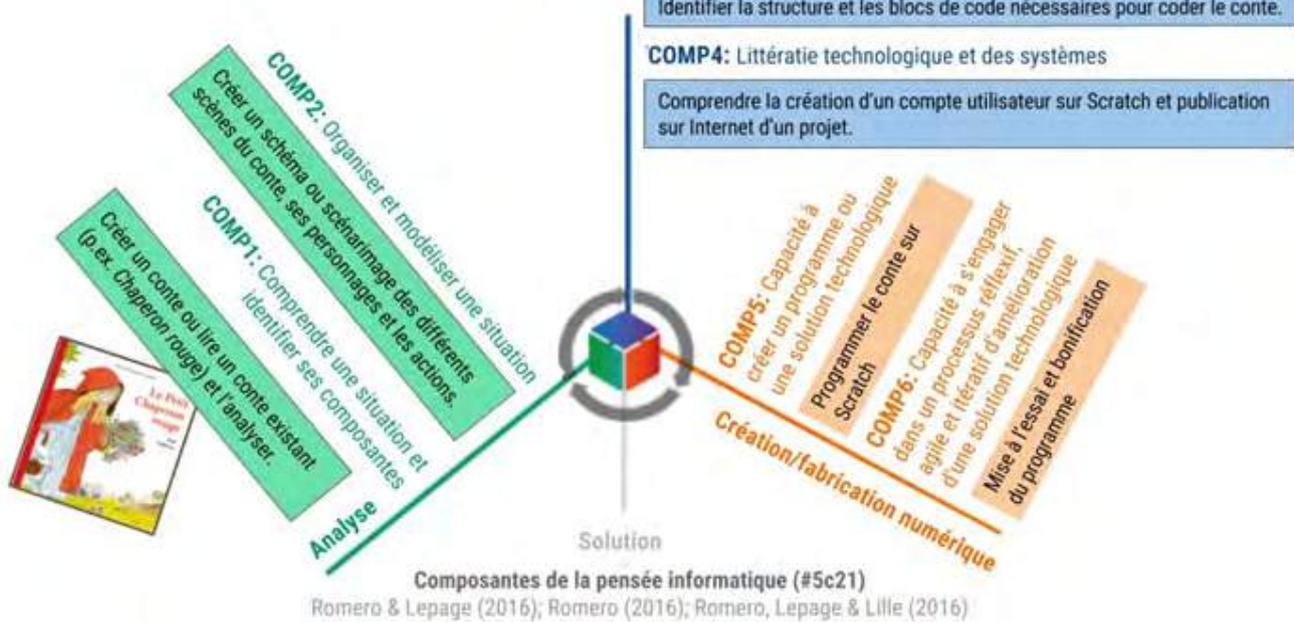
Avant de commencer, nous vous invitons à vous créer un compte Scratch sur le site <http://scratch.mit.edu>

LES DÉFIS DE PROGRAMMATION CRÉATIVE: DE L'HEURE DU CONTE À L'HEURE DU CODE

Les défis de la programmation créative que nous vous proposons dans ce guide permettent développer les composantes de la pensée informatique: de l'analyse, à la littératie numérique à la programmation de la solution. Le schéma ci-dessous présente les 6 étapes de programmation créative (Romero; Lepage; Lille, 2017):

1. La lecture, la compréhension et l'analyse d'un conte ou d'une situation narrative.
2. La modélisation de la situation: la création d'un schéma ou d'un scénarimage des différents scènes du conte, ses personnages et les actions.
3. La littératie du code: l'identification de la structure du programme et les blocs de code nécessaires pour coder le conte.
4. La littératie technologique et des systèmes: création d'un compte utilisateur sur Scratch et gestion du portfolio des projets.
5. La capacité à créer un programme informatique à partir de l'analyse et de la modélisation de la situation.
6. La mise à l'essai et bonification du programme informatique dans une démarche agile et itérative.

De l'heure du conte à l'heure du code. Au cours de cette activité les élèves créent un conte ou choisissent un conte existant, pour ensuite l'analyser, le modéliser et le programmer.



Eh bien, dépôts de toute cette introduction, il est temps de commencer! Bonjour, nous sommes les défenseurs de la pensée informatique et nous vous aiderons à relever tous les défis à venir!

Je m'appelle Cicero et voici Malu, Ana et Bill.



Défi créatif #1a. Vibot et Scratch se présentent



Évaluation : Pensée informatique

- C3 (Littératie en lien au code et aux algorithmes)
- C6 (Approche agile et itérative)

Dans ce premier défi, nous vous invitons à corriger une coquille textuelle qui s'est glissé dans les blocs de programmation Scratch.



Êtes-vous prêts à relever le défi?

Nous vous invitons à vous connecter à Scratch pour rejoindre le projet suivant:



FR: <https://scratch.mit.edu/projects/144235341/>
EN: <https://scratch.mit.edu/projects/144228583/>
PT: <https://scratch.mit.edu/projects/306785156/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)




Dans la partie gauche de l'interface Scratch vous trouverez les blocs de code correspondant à l'animation. Nous vous invitons à modifier le texte ("Vibbbbbbbbot le rrrrrrrobot") pour que Vibot puisse se présenter avec son vrai nom ("Vibot le robot").

Le texte ressemblera à ceci:



- Scratch: "Bonjour!"
- Scratch: "Je m'appelle Scratch! Et toi?"
- Vibot: "Enchanté, Scratch."
- Vibot: "Moi, je m'appelle Vibbbbbbbbot le rrrrrrrobot."
- Vibot: "Oups. J'ai fait une erreur dans mon nom."
- Vibot: "Peux-tu m'aider à corriger mon nom?"

Pour réaliser le défi, nous vous invitons à suivre les deux étapes suivantes:



Étape 1. Analyse de la situation (CTc3).
Où se trouve le bloc de code à modifier?

Étape 2. Corriger et tester la solution (CTc6)
Dans l'éditeur Scratch, nous vous invitons à identifier le bloc de code où le nom de Vibot le robot est mal écrit, afin de le corriger.



Défi créatif #1b. Vibot et Scratch se rencontrent

Nous vous invitons à créer votre premier conte Scratch à partir d'un canevas disponible à l'adresse suivante:



FR: <https://scratch.mit.edu/projects/119727640/>
PT: <https://scratch.mit.edu/projects/306819445/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Votre création doit correspondre au conte ci-dessous.



Vibot et Scratch sont à la cour d'école.
Scratch s'approche de Vibot.

- Scratch: "Salut! Mon nom est Scratch, es-tu un enfant?"
 - Vibot: "Bonjour, mon nom est Vibot, je suis un robot"
 - Scratch: "Je sais communiquer avec des robots, j'aime bien leur donner des ordres et leur faire faire des bêtises."
 - Vibot: "Nous les robots pouvons exécuter certaines instructions"
 - Scratch: "Je connais des instructions qui vont nous permettre faire des bêtises ensemble. Nous allons bien nous entendre ;-)"
 - Vibot: "Tu peux me donner une instruction."
 - Scratch: "Trace une ligne de couleur bleue".
- Après avoir tracé une ligne bleue, Vibot répond à Scratch.
- Vibot: "J'ai tracé une ligne bleue"
 - Scratch: "J'aime bien les robots qui obéissent, je t'aime bien Vibot!"

Avant de commencer à programmer, nous vous invitons à lire attentivement le conte et à répondre aux questions réflexives des questions suivantes:



Étape 1. Analyse de la situation.

- Quels sont les personnages ?
- Quelles sont les actions réalisées par chaque personnage ?
- Où se trouvent les personnages ?

Étape 2. Modélisation de la situation.

Nous vous invitons à créer un schéma ou scénarimage du déroulement du conte. Vous pouvez réaliser le schéma sur papier ou sur un autre support.

Étape 3. Écrire de manière intuitive les algorithmes (p.ex. Quand le programme commence, le personnage Vibot se déplace de 60 pixels à droite).

Nous vous invitons à créer un schéma avant de réaliser votre programmation.



Une fois le conte finalisé vous pouvez l'enregistrer et le partager.

Défi créatif #2a. Vibot cueille un fruit : modélisation



Évaluation : Pensée informatique

- C1 (Analyse)
- C2 (Abstraction et modélisation)

EN : <https://scratch.mit.edu/projects/130395581/>

PT: <https://scratch.mit.edu/projects/306820905/>

Lien vers le document de consignes: <https://goo.gl/szlmJg>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Dans cette histoire, Vibot cueille un fruit. Vous n'avez pas plus d'information sur la situation, mais tout de même, vous devez la représenter afin qu'il soit possible de la réaliser dans Scratch. **À cette étape, ne vous limitez toutefois pas sous prétexte que vous n'êtes pas certains de savoir comment faire dans Scratch.**



Tâche attendue : vous devez identifier tous les objets (lutins) qui vous seraient utiles pour représenter cette situation et identifier des liens qui les unissent. Vous êtes libres de choisir la forme qui vous convient. **N'hésitez pas à faire preuve de créativité, pour autant que la représentation de la situation soit cohérente** (on ne peut pas cueillir de fruits dans une voiture en mouvement, par exemple !).

Vous devez réaliser le tout dans Google Slides en créant un nouveau document: <https://docs.google.com/presentation/u/0/>
Vous devez rendre ce document public afin que nous puissions y avoir accès.



Si vous n'avez pas de compte Google, vous devez vous en créer un:

<https://accounts.google.com/SignUp?hl=fr>

Il vous est possible de cliquer sur "Je préfère utiliser mon adresse e-mail actuelle" pour sauver du temps. Si vous éprouvez des difficultés à cette étape, n'hésitez pas à demander l'aide d'un collègue.

Défi créatif #2b. Vibot cueille un fruit : création



Évaluation : Pensée informatique

- C3 (Algorithmique)
- C5 (Programmation)

Il vous est maintenant demandé de créer l'histoire que vous avez modélisée (défi #1a) dans Scratch. À cette étape, vous pouvez faire des adaptations par rapport à ce que vous avez modélisé préalablement, par exemple pour utiliser les images de la bibliothèque Scratch tout en gardant une histoire cohérente.

Vous devez remixer le projet Scratch suivant dans lequel Vibot a été créé pour vous :

FR:<https://scratch.mit.edu/projects/130549192/>

PT:<https://scratch.mit.edu/projects/306821430/>



Il est primordial de partager votre projet en appuyant sur "Partager" afin de nous le rendre accessible.

La remise se fera via un Google Form dans lequel vous donnerez l'URL de votre projet Scratch :

FR: <https://goo.gl/SZ3slV>

PT: <https://forms.gle/SdG2Lqsy2qCVHSv19>



Défi créatif #3a. Scratch et Vibot ont perdu leurs couleurs



Évaluation :

Aider Scratch et Vibot à retrouver leurs couleurs en utilisant la commande «annuler les effets graphiques» aux scripts des deux personnages. Puis, ajouter des commandes pour faire parler les personnages.

Montrez-leur les liens Bill?

FR: <https://scratch.mit.edu/projects/122835356>
EN: <https://scratch.mit.edu/projects/144236197>
PT: <https://scratch.mit.edu/projects/306825528/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

- Vibot: "Bonjour!"
- Scratch: "Salut!"

Les couleurs de Scratch et Vibot ne cessent de changer.

- Vibot: "Quelqu'un a changé nos couleurs!"
- Scratch: "Tu as raison, Vibot!"
- Scratch: "Pouvez-vous nous aider à retrouver nos couleurs originales?"
- Vibot: "Il faut annuler les effets graphiques."

À toi de jouer!

Les couleurs reviennent à la normale.

- Scratch: "Bien joué!"
- Vibot: "Merci pour ton aide."

Voici
quelques étapes du
défi.



Étape 1. Analyse de la situation (CTc1sc3).

Pourrais-tu identifier les blocs de codes qui font changer les couleurs de Scratch et Vibot?

Étape 2. Modélisation de la situation (CTc4sc2).

Dans la bibliothèque de codes de Scratch, identifier le bloc de code qui va permettre d'annuler les changements de couleurs.

Étape 3. Continuer le dialogue (CTc4sc2)

Dans la bibliothèque de codes de Scratch, sélectionner le bloc de code qui va permettre de programmer le dialogue suivant:

- Scratch: "Bien joué!"
- Vibot: "Merci pour ton aide."

Défi créatif #4a. Vibot et Scratch font de la plongée



FR: <https://scratch.mit.edu/projects/123617273/>
EN: <https://scratch.mit.edu/projects/142553826/>
PT: <https://scratch.mit.edu/projects/306825577/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Reproduire
les dialogues et les
déplacements tels qu'ils le
sont indiqués ci-bas.



Vibot et Scratch font de la plongée sous-marine.
Scratch se déplace à la nage en direction de Vibot

- Scratch: "Yahoo! J'adore nager!"
- Vibot: "Moi aussi."
- Scratch: "Mais où sont tous les poissons?"

Suivez maintenant les étapes ci-dessous.



Étape 1. Analyse de la situation.

- À quel endroit faut-il cliquer pour ajouter un lutin au compte Scratch?
- Quel bloc nous permet de faire déplacer un lutin d'un endroit à l'autre?
- Quel bloc doit-on utiliser pour faire répéter un mouvement plusieurs fois?

Étape 2. Modélisation de la situation.

Dans la bibliothèque de lutins de Scratch, choisir au moins 2 lutins qui savent nager et les placer sur le fond.

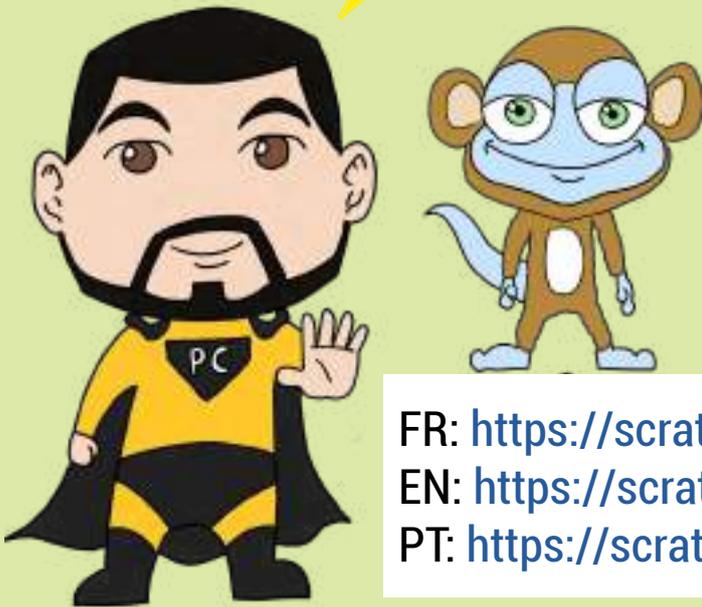
Étape 3. Ajouter des déplacements

À partir de la bibliothèque de Scratch, choisir des blocs de mouvement pour faire déplacer les nouveaux lutins de droite à gauche et de gauche à droite.

Étape 4. Répéter les mouvements

Sélectionner le type de bloc qui permet de répéter les mouvements autant de fois que désiré. Puis, demander aux nouveaux lutins de se déplacer 40 fois.

Défi créatif #5a. Vibot et Scratch aident Petit Singe



FR: <https://scratch.mit.edu/projects/123266307>

EN: <https://scratch.mit.edu/projects/142608034/>

PT: <https://scratch.mit.edu/projects/306825616/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur  !)

- Vibot: "Bonjour."
- Scratch: "Bonjour Petit Singe! Comment vas-tu?"
- Petit Singe: "Je ne vais pas très bien. J'aimerais avoir une banane, mais je n'arrive pas à les atteindre."
- Petit Singe: "Elles sont trop hautes!"
- Scratch: "Nous allons t'aider. N'est-ce pas, Vibot?"
- Vibot: "Certainement."
- Vibot: "Je vais aller voir si je suis assez grand pour les attraper."

Vibot se déplace en direction de l'arbre.

- Vibot: "Malheureusement, je n'arrive pas à les atteindre."

Vibot revient à sa position initiale.

- Scratch: "Je sais!"
- Scratch: "Je vais m'approcher de l'arbre et je vais le secouer pour en faire tomber des bananes!"

Scratch s'approche de l'arbre et le secoue. Puis, il retourne à sa position.

- Scratch: "Ça fonctionne! Elles sont en train de tomber."

À toi de jouer!

Les bananes tombent au sol et Petit Singe se déplace vers elles.

- Petit Singe: "Miam! Quel festin. Merci les amis."

Étape 1 : Répondre aux questions suivantes dans les commentaires du programme

- Décrire l'algorithme qui permet à l'arbre de bouger.
- Quels sont les blocs de codes qui permettent à un personnage ou à un objet de se déplacer d'un endroit à un autre?

Étape 2 : Tâches à réaliser dans le programme

- Corrigez l'erreur suivante : Scratch ne fait pas face à ses amis.
- Scratch doit se déplacer vers l'arbre (mouvement fluide) plutôt que d'y apparaître instantanément.
- Choisir un ou plusieurs blocs qui feront tomber les bananes jusqu'au niveau du sol. Appliquez-vous à rendre le mouvement le plus réaliste possible.
- Le conte doit pouvoir être relancé sans erreur (ex : objets qui restent affichés ou qui ne se replacent pas).
- Dans la bibliothèque de codes de Scratch, ajoutez un bloc grâce auquel Petit Singe pourra remercier ses amis, et insérez-y le texte suivant.

Petit Singe: "Miam! Quel festin. Merci les amis."



Défi créatif #6a. Vibot, Scratch et Gobo jouent au soccer



FR: <https://scratch.mit.edu/projects/123266307>
EN: <https://scratch.mit.edu/projects/142608034/>
PT: <https://scratch.mit.edu/projects/306825616/>

Vibot et Scratch sont avec l'ami de Scratch, Gobo.

- Scratch: "Bonjour, Gobo!"
- Scratch: "Vibot, je te présente mon ami Gobo."
- Vibot: "Bonjour, Gobo. Je m'appelle Vibot. Je suis un robot."
- Gobo: "Bonjour les amis!"
- Gobo: "Voulez-vous jouer au soccer?"
- Scratch: "D'accord!"
- Vibot: "Qu'est-ce que c'est, du soccer?"

À toi de jouer!

Gobo explique à Vibot comment on joue au soccer.

- Gobo: "C'est un sport où on se passe le ballon avec les pieds pour faire des buts."
- Vibot: "D'accord. Jouons!"

Puis, les trois amis se passent le ballon. Le ballon va de Scratch à Gobo à Vibot et l'échange se répète 5 fois.

- Vibot: "J'adore jouer au soccer!"

Étape 1. Analyse de la situation.

- Quels sont les blocs de codes qui permettent à un personnage de se déplacer d'un endroit à un autre?
- Quel bloc de code devons-nous utiliser pour faire parler un personnage?
- Quel bloc nous permet de répéter une action aussi souvent que désiré?

Étape 2. Continuer le dialogue.

Nous vous invitons à choisir, à partir de la bibliothèque de Scratch, le bloc qui permet de faire parler les personnages. Puis, insérer le dialogue suivant:

Gobo explique à Vibot comment on joue au soccer.

- Gobo: "C'est un sport où on se passe le ballon avec les pieds pour faire des buts."
- Vibot: "D'accord. Jouons!"

Étape 3. Contribuer à la situation

Choisir le bloc approprié qui va permettre au ballon de se déplacer d'un joueur à l'autre (Scratch à Gobo, Gobo à Vibot, Vibot à Scratch), selon les coordonnées suivantes:

Scratch. x: -125 y: -95

Gobo. x: 1 y: -49

Vibot. x: 62 y: -116

Quelle forme géométrique est-ce que le déplacement du ballon effectue?
Quelles coordonnées devrait-on attribuer à chaque personnage pour que le triangle soit isocèle? Équilatéral? Scalène?

Étape 4. Répéter les mouvements (CTc1sc9)

Sélectionner le type de bloc qui permet de répéter les mouvements autant de fois que désiré. Puis, programmer le ballon pour qu'il complète la boucle 5 fois.

Étape 5. Terminer le dialogue.

Choisir, le bloc qui permet de faire parler les personnages. Puis, insérer les paroles suivantes:

- Vibot: "J'adore jouer au soccer!"

Défi créatif #7a. Scratch prépare une surprise à Vibot



FR: <https://scratch.mit.edu/projects/123763269/>
EN: <https://scratch.mit.edu/projects/142727239/>
PT: <https://scratch.mit.edu/projects/306825747/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Scratch
prépare une fête surprise pour Vibot.
Dans la pièce décorée se trouvent tous les amis de
Vibot.



- Scratch: "Bonjour!"
- Tous les amis: "Bonjour! Allo! Coucou! Salut! Hello!"
- Scratch: "Nous organisons une fête surprise pour l'anniversaire de Vibot."
- Scratch: "Dans quelques secondes, tous les amis vont disparaître."
- Scratch: "Puis, lorsque Vibot va arriver, les amis vont sortir de leurs"

cachettes."

- Gobo: "J'entends Vibot. Vite, cachons nous!"

Les amis de Vibot disparaissent, mais Scratch reste visible.

Vibot apparaît et se déplace jusqu'à Scratch.

- Scratch: "Vibot!"

Scratch envoie un message ("surprise") à tous les amis pour leur dire qu'il est le temps d'apparaître.

À toi de jouer!

Lorsque les amis reçoivent le message ("surprise") de Scratch, ils se montrent le bout du nez.

- Tous: "Surprise!!"
- Scratch: "Joyeux anniversaire, mon ami!"
- Vibot: "Wow! Merci, les amis. Vous êtes si gentils. Quelle belle surprise!"

Maintenant,
allons aux marches.



Étape 1. Analyse de la situation.

- Quel bloc d'apparence permet de montrer un lutin qui était caché?
- Quel bloc de code contrôle force un lutin à attendre 1+ secondes avant de dire ou de faire quelque chose?
- Comment fait-on pour faire en sorte qu'un lutin dise quelque chose?

Étape 2. Modélisation de la situation.

Choisir le bloc d'apparence qui fera apparaître, lorsqu'ils recevront le message "surprise", les lutins suivants: Giga, Pico, Tera, Gobo, Monkey2, Crab et Nano. Puis, ajouter une commande contrôle pour que les lutins attendent 1 seconde.

Étape 3. Poursuivre le dialogue.

Dans la bibliothèque de codes de Scratch, ajouter un bloc grâce auquel Giga, Pico, Tera, Gobo, Monkey2, Crab, Nano et Scratch pourront surprendre leur ami selon le texte suivant:

- Giga, Pico, Tera, Gobo, Monkey2, Crab, Nano et Scratch: "Surprise!!"
- Scratch: "Joyeux anniversaire, mon ami!"
- Vibot: "Wow! Merci, les amis. Vous êtes si gentils. Quelle belle surprise!"

Défi créatif #8a. Scratch et Vibot perdent leur arrière-plan



FR: <https://scratch.mit.edu/projects/126047102>
EN: <https://scratch.mit.edu/projects/142737514>
PT: <https://scratch.mit.edu/projects/306825816/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Scratch et Vibot sont à la plage et profitent de la belle température.

- Scratch: "Ha!"
- Scratch: "Quelle journée parfaite pour se faire bronzer sur la plage."
- Vibot: "Oui, même si je ne peux pas vraiment bronzer."

Soudainement, les amis se retrouvent sur les pistes de ski.

- Scratch: "Mais que se passe-t-il?"
- Scratch: "Où sommes-nous?"
- Vibot: "Brrr. J'ai froid."
- Vibot: "Les robots ne sont pas faits pour vivre au froid!"

Les amis sont transportés loin des pistes de ski et... dans l'espace!

- Vibot: "Je ne comprends pas ce qui nous arrive!"
- Scratch: "Je crois que quelqu'un essaie de nous jouer un tour."
- Vibot: "En nous envoyant dans l'espace?"

Scratch et Vibot se retrouvent devant un château, au Moyen Âge.

- Vibot: "Et au Moyen Âge?"
- Scratch: "Oui!"
- Scratch: "Quelqu'un a modifié nos arrière-plans."

Un dragon cracheur de feu s'approche dangereusement de Scratch.

- Vibot: "Scratch, attention!"

Les amis disparaissent encore pour réapparaître sur une scène de spectacle devant une foule.

- Scratch: "Oh là là."
- Vibot: "Hum, Scratch. Qu'est-ce qu'on fait?"
- Scratch: "Il faut que quelqu'un remette l'arrière-plan de la plage!!!"

À toi de jouer!

Le micro de Vibot et la guitare de Scratch disparaissent. Les deux amis se retrouvent sur la plage à nouveau, munis de leurs lunettes de soleil.

- Scratch: "Ouff. Quelle aventure!"
- Vibot: "Oui, mais je suis content d'être revenu."
- Scratch: "Moi aussi, merci de nous avoir sortis de ce pétrin."

Étape 1. Analyse de la situation.

- Où doit-on cliquer pour aller gérer les différents arrière-plans?
- Comment peut-on changer un arrière-plan?
- Quel bloc de code événement doit-on utiliser pour basculer d'un arrière-plan à l'autre en recevant un message?
- Quels blocs de code apparence est-ce que l'on doit utiliser pour faire disparaître et faire apparaître différents lutin-objets en passant d'un arrière-plan à l'autre?

Étape 2. Modélisation de la situation.

Dans le script des arrière-plans, ajouter les blocs qui vont permettre à l'arrière-plan de basculer vers la plage lorsque le message "fin spotlight-stage" est envoyé.

Étape 3. Modélisation de la situation.

Dans les scripts des lutins pour le micro et la guitare, ajouter le bloc approprié qui les fera disparaître lorsque l'arrière-plan bascule de la scène à la plage. De plus, il faudra choisir le bloc qui fera apparaître les lunettes et le parasol lorsque l'arrière-plan bascule de la scène à la plage.

Étape 4. Poursuivre le dialogue.

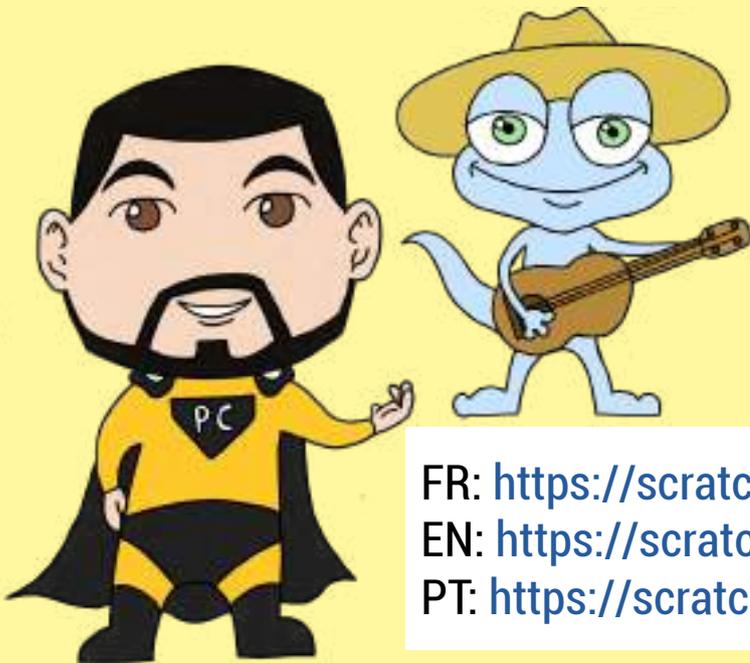
Dans la bibliothèque de codes de Scratch, ajouter un bloc grâce auquel Scratch et Vibot pourront poursuivre le dialogue selon le texte suivant:

- Scratch: "Ouff. Quelle aventure!"
- Vibot: "Oui, mais je suis content d'être revenu."
- Scratch: "Moi aussi, merci de nous avoir sortis de ce pétrin."

Suivre ces étapes facilite les choses, n'est-ce pas?



Défi créatif #9a. Vibot et Scratch forment un groupe de musique



FR: <https://scratch.mit.edu/projects/126847033/>
EN: <https://scratch.mit.edu/projects/142792129>
PT: <https://scratch.mit.edu/projects/306825942/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Je vais vous aider,
voici quelques étapes
pour résoudre ce défi!



Étape 1. Analyse de la situation (CTc3sc2).

- Quelle algorithmes nous permet de faire jouer un instrument lorsqu'un lutin (Scratch, par exemple) touche à l'instrument en question?
- Quel bloc de code permet de rajouter un son à un lutin? Comment peut-on créer une mélodie?

Étape 2. Modélisation de la situation (CTc3sc4).

- Choisir l'algorithme appropriée pour que la guitare électrique se mette à jouer lorsqu'elle est en contact avec Scratch ou Vibot.
- Puis, choisir les sons pour créer une mélodie.

Étape 3. Modélisation de la situation (CTc5sc6).

- Ajouter un 5e instrument et lui faire jouer la mélodie de votre choix par le lutin de votre choix.

Étape 4. Modification de la situation (CTc3sc6).

- Vibot aussi aimerait être capable de jouer de la guitare sèche. Modifier le programme pour faire en sorte que Vibot puisse jouer de la guitare sèche. Comment peut-on faire pour que Scratch et Vibot soient tous les deux capables d'en jouer en même temps?

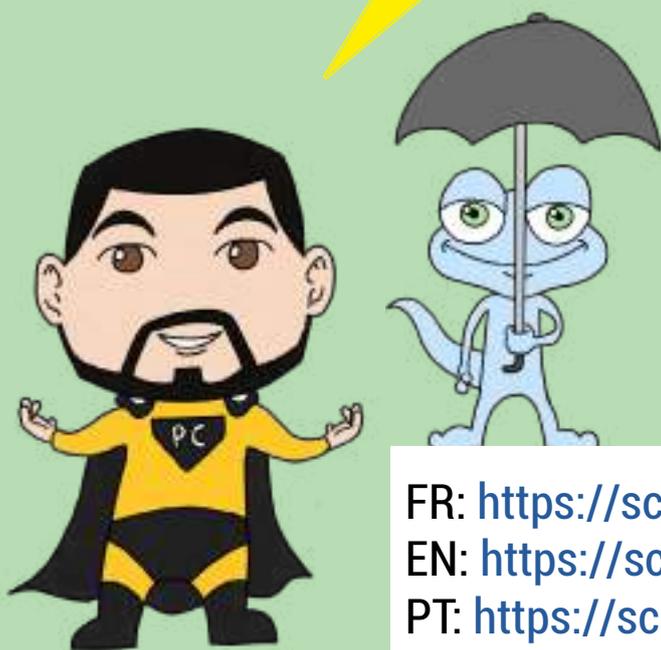
Étape 5. Modification de la situation.

- Créer des chansons avec Scratch et Vibot.
- Si vous souhaitez personnaliser votre groupe, il vous est possible d'enregistrer votre voix et de l'ajouter dans votre projet Scratch. Vous pouvez ensuite utiliser les algorithmes appropriés pour faire en sorte que votre voix se fait entendre lorsqu'un des deux personnages (ou un troisième?) entre en collision avec le microphone qui se trouve au centre de la pièce.

Allons au prochain défi!



Défi créatif #10a. Scratch, Vibot et les saisons



FR: <https://scratch.mit.edu/projects/132948199>

EN: <https://scratch.mit.edu/projects/142785109/>

PT: <https://scratch.mit.edu/projects/306825968/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Ce défi possède 4 arrière-plans, un pour chaque saison de l'année. C'est à vous de déterminer ce qui se passe pendant chaque saison. Chaque saison devrait contenir une courte histoire en lien avec la saison. Par contre, vous devez répondre à certaines contraintes.



Étape 1. Modélisation

Avant de commencer à programmer, organisez vos idées de façon logique en les écrivant sur une feuille de papier ou dans un document Google Docs. Sur ce document, inscrivez les éléments (lutins, blocs de codes, algorithmes, dialogues, etc.) Scratch dont vous aurez besoin pour chaque saison, en tenant compte des contraintes.

Étape 2. Contraintes

- Printemps: Au printemps, les fleurs poussent. Ajouter une fleur qui pousse progressivement (autrement dit, la fleur ne devrait pas apparaître d'un moment à l'autre et devrait plutôt pousser sous nos yeux).
- Été: Ajoutez au moins 2 autres personnages pour tenir compagnie à Scratch et Vibot.
- Automne: Insérez des feuilles d'arbre qui tombent en continu.
- Hiver: Puisque c'est l'hiver, vous devez ajouter des flocons de neige qui tombent de façon continue. De plus, Scratch et Vibot construisent un bonhomme de neige. Assurez-vous qu'il y ait une progression dans la construction du bonhomme de neige (autrement dit, le bonhomme de neige ne devrait pas simplement apparaître d'un moment à l'autre).
- Pour ajouter du réalisme, assurez-vous que les personnages soient habillés de façon appropriée à la saison dans laquelle ils se trouvent.

Étape 3. Inclure des transitions

Pour chaque saison, il y a un lutin avec le nom de la saison formé de lettres colorées. Utilisez les blocs de codes qui vous permettront de faire apparaître le noms des saisons aussitôt que vous arrivez sur l'arrière-plan approprié. Bien que des effets visuels aient déjà été appliqués aux noms des saisons, vous pouvez les changer pour ajouter les effets de votre choix.

Étape 4. Soyez créatif!

Laissez aller votre créativité Malgré les restrictions, vous pouvez faire ce que vous voulez avec chaque saison. Par exemple, une saison pourrait être un jeu alors qu'une autre saison serait une histoire. Amusez-vous!

Défi créatif #11a. Scratch passe l'Halloween



FR: <https://scratch.mit.edu/projects/128340027>
EN: <https://scratch.mit.edu/projects/144254480>
PT: <https://scratch.mit.edu/projects/306826007/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Étape 1. Analyse de la situation (CTc3sc2)

Sur une feuille de papier, répondez aux questions suivantes et expliquez, dans vos mots, ce qui se produit dans chaque situation:

- Quelle algorithme permet de faire déplacer aléatoirement un lutin? Comment peut-on lui faire faire un déplacement précis en boucle (par exemple, lui faire faire un carré)?
- Quelle algorithme doit-on utiliser pour créer un effet lorsqu'il y a une collision (deux lutins qui se touchent) entre deux lutins?

Étape 2. Compléter le programme (CTc3sc4)

- Choisir l'algorithme appropriée pour faire bouger la chauve-souris selon une trajectoire de votre choix (forme géométrique).
- Assurez-vous de faire perdre un point à Scratch lorsqu'il y a une collision entre lui et la chauve-souris.

Étape 3. Corriger le programme (CTc6sc1 et CTc6sc2)

- Il y a une erreur dans le programme du niveau deux. Identifiez et corrigez cette erreur pour qu'on puisse jouer au jeu sans problème.

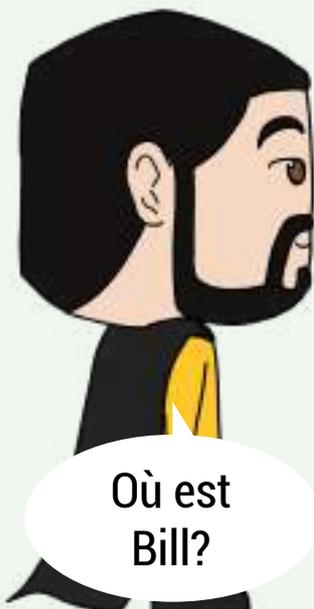
Étape 4. Modifier le programme (CTc5sc6)

- Trouvez une façon de rendre le jeu (niveaux 1 et 2) plus difficile.

Étape 5. Agrandir le programme (CTc5)

- Ajoutez un niveau 3 au programme. Trouvez un arrière-plan relié à la thématique et donnez-lui un nom approprié.
- Assurez-vous que la transition entre le niveau 2 et le niveau 3 soit fluide et efficace.
- Ajoutez un troisième obstacle, en conservant le fantôme et la chauve-

Défi créatif #12a. Scratch, Vibot, et le jeu d'évasion



Où est
Bill?

FR: <https://scratch.mit.edu/projects/128340027>
EN: <https://scratch.mit.edu/projects/144254480>
PT: <https://scratch.mit.edu/projects/306826007/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Étape 1. Analyse de la situation

- Quel est le but du jeu, et quel différents systèmes nous permet de réussir?
- Est-ce qu'il est possible de trouver les indices sans cliquer partout?

Étape 2. Comprendre la situation

- Comment est-ce peut-on définir ce qu'est un code?
- Est-ce qu'un code peut être formulé de différentes façons (programmation, lettres, chiffres, couleurs, symboles, etc.)?
- Comment est-ce qu'un certain code peut mener à la réalisation d'une certaine action?
- Dans ce jeu, il y a une hiérarchie importante dans les différents événements qui peuvent se produire (par exemple, si on clique sur A, B apparaît). Comment expliquer cette hiérarchie en langage normal et en langage Scratch?
- Est-ce qu'il y a une ou plusieurs collisions entre deux lutins? Si oui, quelles sont ses ou leurs fonctions?

Étape 3. Corriger le script

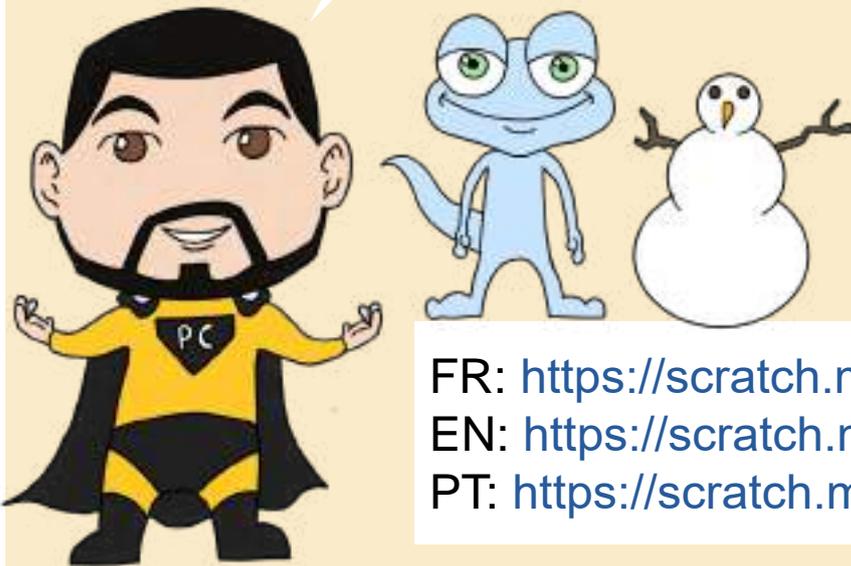
- Il y a une incohérence dans la pièce 2, pouvez-vous la trouver? Pouvez-vous la corriger de façon à ce que le mouvement escompté soit aussi naturel que possible?

Étape 4. Compléter le programme

- Complétez le programme pour créer un autre système de code pour que Scratch et Vibot puissent sortir de la 2e pièce.
- Ajoutez des lutins et cachez vos indices dans la pièce.
- Créez votre propre jeu d'évasion.

Défi créatif #13a. Scratch et Vibot fuient les balles de neige

En vous basant sur le défi créatif #11a, créez un jeu dans lequel Scratch doit se sauver de Pingouin Coquin et de ses balles de neige. Trouvez une façon pour que Scratch puisse faire des points (suggestions: collision avec Vibot, attraper des objets de votre choix qui se déplacent aléatoirement, ou simplement éviter un certain nombre de balles de neige envoyées par Pingouin Coquin. Soyez créatifs!)



FR: <https://scratch.mit.edu/projects/128340027>
EN: <https://scratch.mit.edu/projects/144254480>
PT: <https://scratch.mit.edu/projects/306826007/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Étape 1. Analyse de la situation (CTc3sc2)

Quelle algorithmes permet de faire déplacer aléatoirement un lutin? Comment peut-on lui faire faire un déplacement précis en boucle (par exemple, lui faire faire un carré)?

- Quelle algorithmes doit-on utiliser pour créer un effet lorsqu'il y a une collision (deux lutins qui se touchent) entre deux lutins?
- Comment installer un système de points qui déterminera si Scratch perd ou gagne la partie?

Étape 2. Modéliser la situation

Sur une feuille de papier ou un document Google Docs, inscrire les différents systèmes dont vous aurez besoin pour réussir le jeu (algorithmes de mouvement, fonction “broadcast” pour qu’un lutin envoie un message à un autre lutin, système de points, etc.)

- Lorsque vos systèmes auront clairement été analysés et modélisés, vous pouvez passer à l’étape de la programmation.

Étape 3. Compléter le programme (CTc3sc4)

Trouvez les algorithmes qui permettront aux trois lutins de se déplacer d’un endroit à l’autre (Scratch se déplacera en réponse à une commande de la part du joueur et Pingouin Coquin et Vibot se déplaceront soit aléatoirement ou selon une trajectoire que vous aurez prédéfinie).

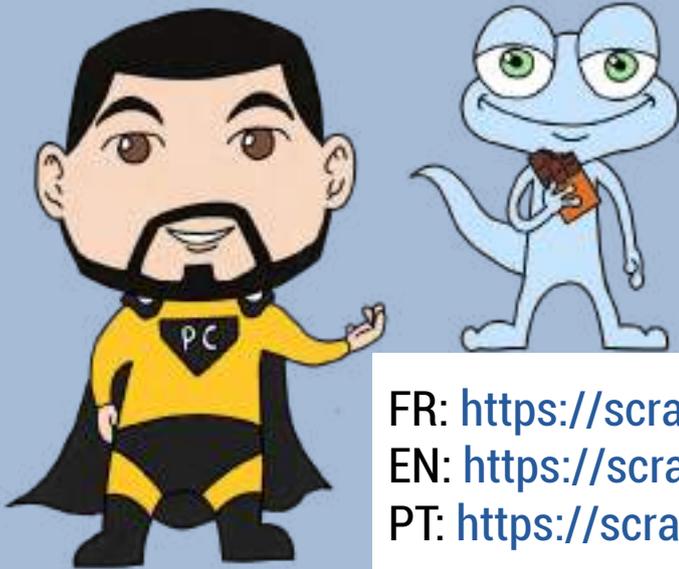
- Trouvez une façon pour que Pingouin puisse envoyer des balles de neige en direction de ses amis. Lorsqu’il y aura collision entre une balle de neige et Scratch, Scratch perdra un point.
- Établissez un système de points. Déterminez le nombre de points nécessaires pour gagner et, à l’inverse, la quantité de balles de neige qui fera perdre Scratch.
- Assurez vous que les mouvements des lutins soient fluides et naturels. N’oubliez pas que le programme doit pouvoir être relancé sans problème.

Suggestions:

Vous pouvez ajouter différentes difficultés si vous considérez votre jeu trop facile.

- Il vous sera possible de créer plusieurs niveaux, en intégrant des nouveaux obstacles dans chaque niveau.
- Vous pouvez ajouter un abris (par exemple un arbre ou une montagne de neige) en arrière de laquelle Scratch sera en sécurité pendant une certaine période de temps.
- Pourquoi ne pas ajouter une mélodie question d’améliorer l’ambiance de votre jeu?

Défi créatif #14a. Vibot et les chocolats



FR: <https://scratch.mit.edu/projects/133059181/>
EN: <https://scratch.mit.edu/projects/146235612/>
PT: <https://scratch.mit.edu/projects/306826070/>

Dans ce défi, Vibot part à la chasse aux oeufs de Pâques! Plusieurs oeufs ont été cachés dans un endroit de votre choix (votre cours d'école, votre maison, votre salle de cours, etc.). C'est à vous de déterminer comment se déroule la scène et d'aider Vibot à trouver les chocolats dont il raffole tant.



Étape 1. Analyse de la situation

- Comment est-ce que l'on peut ajouter un arrière-plan de notre choix sur Scratch?
- De quelle façon peut-on avoir des objets cachés, et quels blocs nous permettent de les faire apparaître?
- Comment peut-on créer nos propres lutins?
- Quels blocs de codes nous permettent de créer une succession d'évènements? (Par exemple, si Vibot touche A, A envoie un message à B et C, qui à leurs tours effectuent un mouvement quelconque).

Étape 2. Modélisation

- Sur un document Google Docs, identifiez les différents éléments (arrière-plan, lutins, blocs de codes, dialogues, etc.) dont vous aurez besoin pour réaliser votre défi.
- Il peut vous être utile de dessiner un croquis, sur une feuille de papier, de la scène que vous aimeriez créer. Vous pourriez ainsi déterminer à l'avance les endroits où vous pouvez cacher vos oeufs.
- Quelle est l'importance d'une telle préparation avant de commencer à programmer?

Étape 3. Choisir un arrière-plan

- Choisir un endroit de votre quotidien qui vous intéresse. Cet endroit peut être votre salle de cours, votre maison, la cour de l'école, votre rue, etc. Assurez-vous que le décor est assez vaste pour que Vibot puisse s'y déplacer pour trouver les oeufs.
- Lorsque votre endroit idéal aura été choisi, prenez-le en photo et importez la photo sur Scratch.

Étape 4. Création et insertion de lutins

- Créez vos propres lutin-oeufs et colorez-les de la façon de votre choix.
- Ensuite, trouvez une façon pour dissimuler les oeufs dans le décor. Ceci peut être réalisé de plusieurs façons différentes. Plus vous cachez les

oeufs de différentes façons, mieux c'est. Soyez créatif!

- Vous pouvez inclure d'autres lutins dans votre défi, comme par exemple des objets qui contribueront au décor, ou des personnages qui aideront Vibot dans sa chasse.

Étape 5. Création d'un scénario complet

- Déterminez de quelle façon se déroulera la chasse aux oeufs de Pâques de Vibot: est-ce que ce sera sous forme de jeu, dans lequel le joueur devra manipuler Vibot pour l'aider à trouver ses oeufs, ou est-ce que ce sera sous forme d'histoire, où chaque élément du défi aura été programmé par vous à l'avance? N'hésitez pas à vous inspirer des autres défis présents dans ce guide.
- Pensez à intégrer des dialogues dans votre défi.

Variations

- Si vous songez à créer un jeu, il serait intéressant d'utiliser un système de points (blocs de codes "Données" dans Scratch)
- Si vous vous sentez assez à l'aise, il serait possible pour vous d'ajouter plusieurs scènes au défi. Ainsi, vous pourrez prendre quelques photos qui représentent votre quotidien, et cacher les oeufs de chocolat dans les différentes scènes.
- Scratch permet de inclure des sons et de la musique dans ses projets. Il serait donc possible d'ajouter un son de victoire à chaque fois que Vibot trouve un oeuf. Vous pourriez également mettre une chanson de votre choix, question de donner un aperçu encore plus juste de votre quotidien.

Soyez créatif!



Défi créatif #15a. Vibot et les engrenages de sa voiture



Ce défi est pour
vous...
Essayez de construire le projet et
de partager le lien!

Défi créatif #16a. Scratch décrit Noël à Vibot

Dans ce défi, Vibot et Scratch jouaient dans la neige. Soudainement, un objet est tombé du ciel. Vibot pense que c'est un objet mais c'est plutôt Papa Noël. Scratch essaye de décrire ce qui c'est la Noël à Vibot. Vous allez continuer l'histoire et le coder dans Scratch.

Essayez de construire le projet et de
partager le lien!



Scratch et Vibot jouaient dans la neige.

- Scratch: "Huhu Scratch: "Regarde Vibot, j'ai créer un téléphone avec la neige ."
- Vibot: "Ohh, c'est cool!." Moi, j'ai créer une voiture

Soudainement, quelque chose tombe du ciel devant Vibot.

- Vibot: "Viens voir Scratch, un objet est tombé du ciel?"

Scratch se deplace vers Vibot

- Scratch: "Hoo, c'est Papa Noel!"
- Vibot: "C'est quoi Papa Noel et c'est quoi Noël."

Continuer le dialogue en décrivant à Vibot ce que c'est la Noël et Papa Noël à travers le personnage Scratch.

En bas, voici les étapes à suivre.



Étape 1. Analyse de la situation (CTc3sc2)

- Comment est-ce que l'on peut ajouter un arrière-plan de la neige sur Scratch?
- De quelle façon peut-on faire tomber des objets du ciel?
- Quelle sont les lutins qu'on a besoin pour programmer ce défis?
- Quels blocs de codes nous permettent de créer une succession d'évènements?

Étape 2. Modélisation

- Sur un document Google Docs, identifiez les différents éléments (arrière-plan, lutins, blocs de codes, dialogues, etc.) dont vous aurez besoin pour réaliser votre défi.
- Il peut vous être utile de chercher des arrière-plans et des images de Papa Noël sur l'internet.
- Quelle est l'importance d'une telle préparation avant de commencer à programmer?

Étape 3. Choisissez un arrière-plan / arrière-plan

- Choisissez un paysage de Noël / photos qui vous intéresse.
- Une fois l'emplacement idéal choisi, entrez-le dans Scratch.

Étape 4. Création et insertion de caractères

- Créez vos propres personnages en plus du père Noël et vous pourrez les colorier à votre guise.
- Soyez créatif!

Étape 5. Création d'un scénario complet

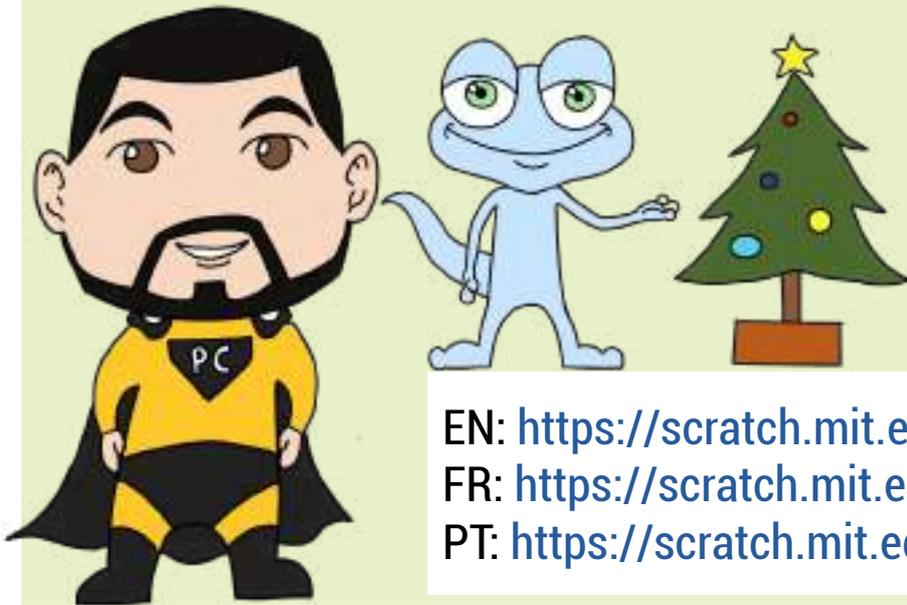
- Déterminez de quelle façon Scratch va décrire Noël à Vibot? N'hésitez pas à vous inspirer des autres défis présents dans ce guide.
- Pensez à intégrer des dialogues dans votre défi.

Maintenant, passons à un autre défi.



Défi créatif #16b. Vibot et Scratch célèbrent Noël

Dans ce défi, Vibot et Scratch se préparent à célébrer Noël. Ils commenceront par décorer la pièce, puis ils devront emballer deux cadeaux.



EN: <https://scratch.mit.edu/projects/146251411/>
FR: <https://scratch.mit.edu/projects/137697840/>
PT: <https://scratch.mit.edu/projects/306826152/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Étape 1. Analyse de la situation (CTc3sc2)

- De quel type de pièce s'agit-il?
- Quel est le thème de la scène dans laquelle Scratch et Vibot se trouvent? Pouvons-nous ajouter des éléments qui correspondent au thème?
- Il y a un objet qui ne correspond pas au thème; peux-tu l'identifier? Si oui, pouvez-vous faire en sorte que l'objet devienne en lien avec le thème, sans le faire disparaître? Soyez créatifs!

Étape 2. Ajout d'éléments festifs

- Scratch et Vibot célèbrent Noël: ajoutez-leurs des habits appropriés!
- Que pourrait-on placer au dessus de la cheminée?
- Ajoutez des décorations dans la pièce, aux endroits de votre choix.

Étape 3. Animez la source de chaleur

- Il y a des flammes dans la cheminée, mais celles-ci sont immobiles. Trouvez une façon de faire en sorte que les flammes soient animées de façon naturelle.

Étape 4. Décorez le sapin!

- En vous servant des différents articles se trouvant dans les boîtes, aidez Scratch et Vibot à décorer le sapin de Noël.
- Vous pouvez ajouter des articles qui ne se trouvent pas dans les boîtes. Soyez créatifs; il faut que l'arbre soit à votre goût!
- Il manque un objet pour placer sur la cîme de l'arbre...

Étape 5. Emballez les cadeaux

- Il y a deux boîtes à la droite de la pièce; emballez-les et décorez-les.

Étape 6. Créez de l'ambiance!

- Ajoutez de la musique de circonstance, question de créer une belle ambiance pour Scratch et Vibot.
- Pourquoi ne pas ajouter un bruit de feu de foyer?

Étape 7. Créez une belle histoire.

Créez un scénario dans lequel Scratch et Vibot s'offrent des cadeaux de Noël. Votre scénario devra comprendre:

- Des dialogues.
- Des mouvements réalistes (déplacements, un ami qui donne et un ami qui reçoit un cadeau, un ami qui déballe une boîte et qui en sort le cadeau).

Défi créatif #17a. Scratch et Vibot vont au Carnaval de Québec



EN: <https://scratch.mit.edu/projects/146251411/>
FR: <https://scratch.mit.edu/projects/137697840/>
PT: <https://scratch.mit.edu/projects/306826152/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Dans ce défi, Scratch et Vibot se rendent au Carnaval de Québec où ils trouvent plusieurs activités à faire. Le Carnaval leur offre la possibilité de faire de la glissade sur neige, patiner, construire un palais de glace, manger de la tige d'érable et faire des anges dans la neige. Bien que chaque activité ait été «installée» dans le projet Scratch, c'est à vous d'ajouter les éléments essentiels à la réussite des activités. Suivez les instructions pour offrir à Vibot et Scratch un Carnaval inoubliable!



Activité 1. Glissade sur neige

- Vibot et Scratch souhaitent glisser, mais ils n'ont pas de toboggan. À vous de leur en bâtir un!
- Ensuite, ajoutez les blocs de mouvement nécessaires à leur descente. Assurez-vous que le mouvement de descente soit naturel.

Activité 2. Patinage et hockey sur glace

- On vous demande de réparer quelques erreurs de programmation. La première erreur se produit lorsque Pico patine sur la neige, ce qui n'est pas réaliste. La deuxième est lorsque Pico patine à reculons, ce qui n'est pas un mouvement très naturel.
- Pour ajouter du réalisme au défi, assurez-vous que le corps de Pico se tourne vers la bonne direction lorsqu'il patine.
- De plus, vous pouvez ajouter des personnage (2 à 4) et les faire jouer une partie de hockey. Ils auront besoin d'instruments (bâtons, rondelle, buts, etc.). Assurez-vous d'inclure le mouvement des bâtons et un déplacement à la rondelle.

Activité 3. Construction d'un palais de glace

- Dans cette activité, on vous demande de construire un palais de glace en vous servant d'un bloc de glace qui est disponible (indice: vous pouvez dupliquer le bloc pour en avoir plus qu'un. N'hésitez pas à régler la taille des blocs au goût!).

Activité 4. Tire d'érable sur neige

- Vous avez à votre disposition une chaudière dans laquelle se trouve une délicieuse tire d'érable. À vous de la faire couler (encore une fois, on demande que le mouvement soit naturel) et d'ajouter des mouvements au bâton de bois pour qu'il puisse aller récupérer la tire.

Activité 5. Anges dans la neige

- Dans cette activité fort simple, vous devez ajouter des mouvements à Scratch pour qu'il puisse dessiner un ange dans la neige. Évidemment, les mouvements doivent modifier, d'une certaine façon, la neige sur laquelle Scratch est étendu.

Activité 6. Activité secrète et cachée

- À vous de la trouver...

Défi créatif #18a. Vibot, Scratch et le sport national canadien

Dans ce défi, Vibot, Scratch et leurs amis font partie d'un jeu de hockey sur table. Vous êtes l'équipe bleue et contrôlez Scratch (WASD), Pico (flèches) et Nano, le gardien de but (barre d'espace).



EN: <https://scratch.mit.edu/projects/146655561/>
FR: <https://scratch.mit.edu/projects/136979543/>
PT: <https://scratch.mit.edu/projects/306826337/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Étape 1. Jouez!

- Testez le jeu. Pour gagner, il faut «scorer» 5 buts.
- Essayez d'identifier les principaux problèmes ou «bugs» que le programme contient.

Étape 2. Modélisation

- Sur une feuille de papier, faire une liste des problèmes à régler.
- Pour chaque problème, expliquez dans vos mots la façon dont vous allez vous prendre pour régler le problème.
- Ensuite, expliquer en langage Scratch (c'est-à-dire en expliquant quels blocs et algorithmes seront à utiliser) la façon dont vous allez vous prendre pour régler le problème.
- Lorsque les problèmes et les solutions seront bien identifiés, commencez à faire des modifications au programme.

Étape 3. Modifiez le programme

- Quatre personnages ne bougent pas. En vous inspirant des autres personnages, donnez des mouvements à Vibot, Snowman, Dino et M. Créature. Assurez-vous que les personnages ne puissent pas sortir de leurs «lignes».
- Ajoutez une introduction au programme (par exemple, une introduction qui inclut un bouton «Démarrer».
- Il faut cinq points pour gagner la partie. Ajouter un algorithme qui fait en sorte que le jeu s'arrête lorsque le joueur ou l'ordinateur marque 5 points.
- La transition entre un but et le retour au jeu n'est pas naturelle. Assurez-vous de modifier cette transition de façon à ce qu'il y ait un petit arrêt entre le but et le retour au jeu.
- D'autres erreurs se sont glissées. Réglez-les de façon à ce que le programme se déroule bien.

Défi créatif #18b. Vibot, Scratch et le sport national canadien (à vous de programmer!)



Crie o seu próprio jogo de hóquei de mesa.



EN: <https://scratch.mit.edu/projects/146655561/>
FR: <https://scratch.mit.edu/projects/136979543/>
PT: <https://scratch.mit.edu/projects/306826337/>



(* Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Créez votre propre jeu de hockey sur table

- Créez un jeu dans le genre «Air hockey». L'arrière-plan est déjà dans le programme, mais vous pouvez créer un nouveau document Scratch pour repartir à neuf, ou modifier l'arrière plan qui est déjà là.
- Le jeu aura une rondelle et deux maillets avec lesquels on peut marquer des buts.
- Votre jeu doit comprendre quelques restrictions:
- Les maillets ne peuvent pas aller plus loin que la ligne du milieu
- La rondelle doit pouvoir rebondir lorsqu'elle entre en collision avec les maillets ou la bande.
- La vitesse de la rondelle doit diminuer graduellement lorsqu'elle n'entre pas en collision avec la bande ou les maillets
- Le maillet du joueur doit être contrôlé par la souris. (Variation: Vous pouvez créer un jeu où deux joueurs peuvent s'affronter. Dans ce cas, il vous faudrait utiliser les commandes sur le clavier (WASD et flèches) pour faire bouger les maillets).

- Le maillet adverse (celui de l'ordinateur) doit pouvoir se déplacer vers la rondelle pour tenter de la frapper.
- Pour les mouvements de la rondelle et des maillets, n'hésitez pas à vous inspirer du défi #18a.
- Vous pouvez créer plusieurs niveaux de difficultés. Il y a plusieurs façons d'y arriver (indice: vitesse de la rondelle).

Desafio criativo #19a. O Vibot em Londres



Dans ce court défi, Vibot veut visiter Londres. Pour ce faire, vous devez l'aider à embarquer dans l'autobus et à faire le faire avancer.

FR / EN: <https://scratch.mit.edu/projects/122831125/>
 PT: <https://scratch.mit.edu/projects/306826368/>

(*  Avant de commencer la programmation du défi, n'oubliez pas de remixer le projet en cliquant sur [Remix](#) !)

Étape 1. Aidez Vibot à embarquer dans l'autobus

- Quels blocs de codes permettent à un lutin de se déplacer naturellement d'un point à l'autre?
- Utilisez les codes qui permettront à Vibot de se déplacer de sa position initiale à l'autobus et à y entrer.
- Assurez-vous que le déplacement de Vibot soit le plus naturel possible (par exemple, qu'est-ce qui se produit lorsque quelqu'un s'éloigne de nous? Est-ce que cette personne conserve la même taille?)
- Comment peut-on faire en sorte que l'on puisse voir Vibot seulement qu'à partir de les fenêtres de l'autobus?

- Ajoutez d'autres lutins-passagers dans l'autobus. Ou encore mieux, prenez un «selfie» et allez faire du tourisme avec Vibot en ajoutant votre photo sur Scratch!

Étape 2. Faire avancer l'autobus

- Quels codes doit-on utiliser pour faire en sorte que l'autobus se mette en mouvement dès que Vibot ait pris place dans l'autobus? Quel algorithme doit-on utiliser pour créer un effet lorsqu'il y a une collision (deux lutins qui se touchent) entre deux lutins? Est-ce qu'il y a une façon d'ajouter un délais entre la collision et la réaction?
- Faites déplacer l'autobus de son endroit initial jusqu'à l'autre bout de l'image.
- Assurez-vous que le déplacement de l'autobus ne soit pas trop rapide et soit naturel. Par exemple, pouvez-vous faire en sorte que les roues de l'autobus tournent pendant le déplacement?

Étape 3. Soyez créatifs!

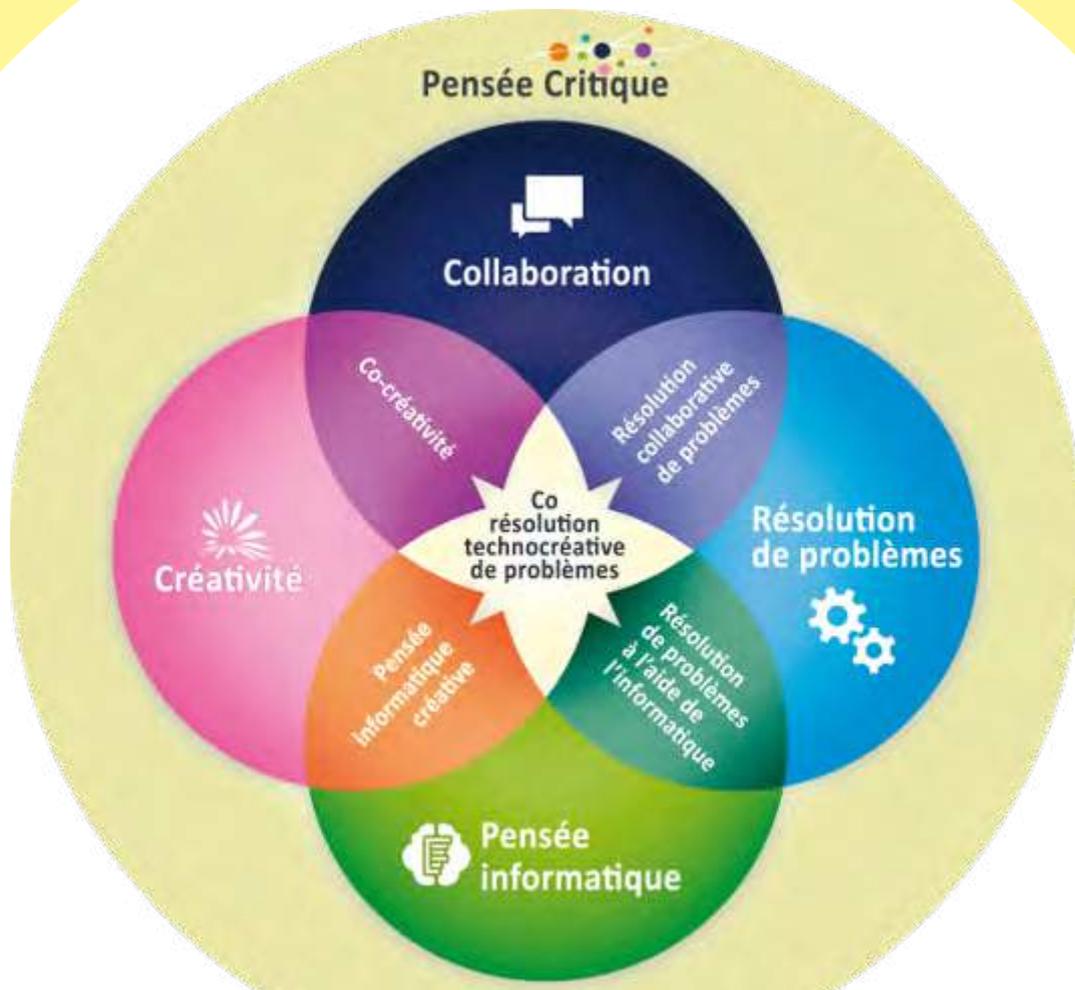
- Amusez-vous! Vous pouvez ajouter d'autres lutins, objets ou paysages pour prolonger le voyage de Vibot.





Pistes pour l'évaluation des défis selon les Compétences pour le siècle 21(21)

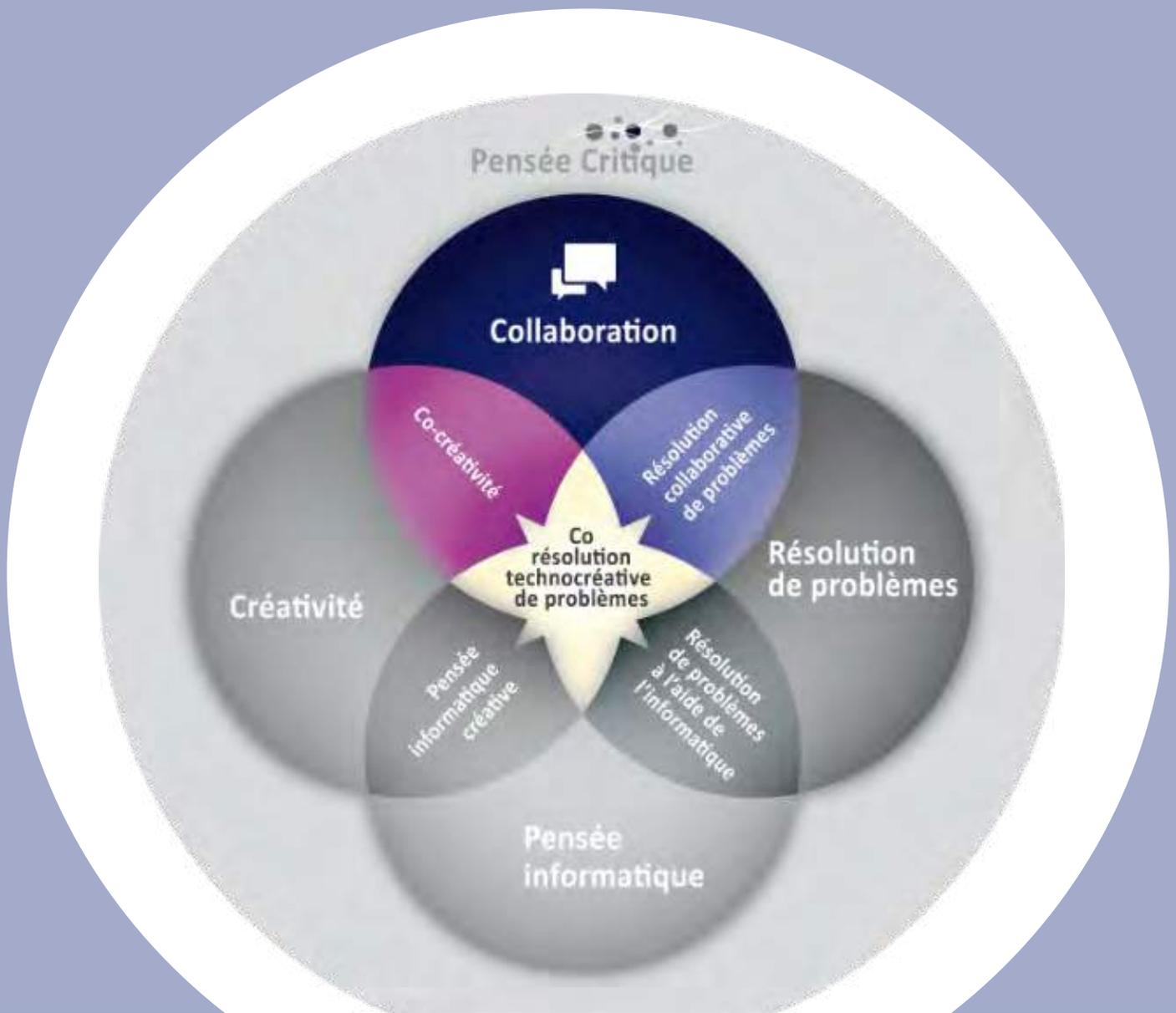
Composants des Compétences pour le siècle 21:



La **pensée critique (PC)** est la capacité de développer une réflexion critique indépendante. La pensée critique permet l'analyse des idées, des connaissances et des processus en lien avec un système de valeurs et de jugements propre. C'est une pensée responsable qui s'appuie sur des critères et qui est sensible au contexte et aux autres.

- **Composant 1 (PC-c1):** Cerner les composants d'une idée ou d'une œuvre.
- **Composant 2 (PC-c2):** Explorer les différentes perspectives et positionnements en lien à une idée ou une œuvre.
- **Composant 3 (PC-c3):** Se positionner par rapport à une idée ou à une œuvre.





La **collaboration (CO)** est la capacité de développer une compréhension partagée et de travailler de manière coordonnée avec plusieurs personnes dans un objectif commun.

- **Composant 1 (CO-c1):** Identifier la situation problème et définir un objectif commun.
- **Composant 2 (CO-c2):** Établir et maintenir une compréhension et une organisation partagée.
- **Composant 3 (CO-c3):** Développer une compréhension des savoirs, compétences, forces et limitations des autres membres de l'équipe pour organiser les tâches envers l'objectif commun.
- **Composant 4 (CO-c4):** Savoir gérer les difficultés du travail en équipe dans le respect et la recherche de solutions.
- **Composant 5 (CO-c5):** (Co)construction de connaissances et/ou artefacts





La **créativité (CREA)** est un processus de conception d'une solution jugée nouvelle, innovante et pertinente pour répondre à une situation-problème.

➤ **Composant 1 (Crea-c1): Incubation des idées**

- Crea-c1-sc01: Explorer une variété de démarches ou de solutions nouvelles.
- Crea-c1-sc02: Incubation de différentes opportunités, stratégies et concepts

➤ **Composant 2 (Crea-c2): Génération d'idées**

- Crea-c2-sc01: Génération d'idées, brainstorming, exploration divergente
- Crea-c2-sc02: Utiliser des sources d'inspiration pour orienter la recherche créative.
- Crea-c2-sc03: Combinaison d'idées

➤ **Composant 3 (Crea-c3): Évaluation et sélection**

- Crea-c3-sc01: Relevance et pertinence par rapport à la situation-problème
- Crea-c3-sc02: Nouveauté et originalité
- Crea-c3-sc03: Élégance et parsimoni
- Crea-c3-sc04: Capacité de genèse prospective





La **pensée informatique (PI)** est un ensemble de stratégies cognitives et métacognitives liées à la modélisation de connaissances et de processus, à l'abstraction, à l'algorithmique, à l'identification, à la décomposition ainsi qu'à l'organisation de structures complexes et de suites logiques.

➤ **Composant 1 (CTc1): Analyse**

- CTc1sc01: Comprendre/cerner une situation-problème
- CTc1sc02: Adopter une perspective (p. ex. : de quelqu'un d'autre, d'une machine ou d'un objet)
- CTc1sc03: Identifier les objets clés d'une situation
- CTc1sc04: Identifier des attributs de chaque objet et leur changement dans le temps (si dynamiques)
- CTc1sc05: Identifier des ressemblances ou des différences entre les objets
- CTc1sc06: Identifier des événements et ce qu'ils déclenchent
- CTc1sc07: Identifier des relations entre les entités (p. ex. : dépendance, hiérarchie, causalité)

➤ **Composant 2 (CTc2): Modélisation**

- CTc2sc01: Prendre conscience de l'importance de planifier une solution avant de la créer
- CTc2sc02: Choisir un outil approprié pour exprimer un modèle d'un don-

nées et ses relations

- CTc2sc03: Produire une représentation (modélisation) d'une situation
- CTc2sc04: Optimiser l'organisation des données (p.ex. éviter les répétitions)

➤ **Composant 3 (CTc3): Code litteraty**

- CTc3sc01: Décrire un algorithme en mots ou à partir de pseudo-code
- CTc3sc02: Expliquer le sens d'un algorithme existant
- CTc3sc03: Comprendre et identifier les concepts de la programmation :

blocs, objets, instructions et opérateurs

- CTc3sc04: Apporter des modifications à un algorithme existant
- CTc3sc05: Concevoir un programme à partir de l'analyse/modélisation

d'une situation

➤ **Composant 4 (CTc4): Systèmes**

- CTc4sc01: Comprendre les fonctionnalités et composantes de différentes technologies (réseau, ordinateur, robot...)

○ CTc4sc02: Choisir d'une ou plusieurs technologies adaptées à l'analyse/modélisation de la situation

- CTc4sc03: Distinguer les logiciels informatiques et le matériel informatique

○ CTc4sc04: Identifier les limites d'un logiciel ou d'un appareil informatique

- CTc4sc05: Décrire de manière cohérente le fonctionnement d'un système

complexe

➤ **Composant 5 (CTc5): Créer le programme**

- CTc5sc01: Choisir un langage approprié à la situation ou à la technologie (robot, tablette)

○ CTc5sc02: Décomposer un objectif en une séquence d'instructions suffisamment précises pour le langage informatique utilisé

- CTc5sc03: Identifier ou écrire des fonctions ou blocs de code associés à des résultats (suite à l'exécution d'un programme)

○ CTc5sc04: Programmée en utilisant les techniques offertes par la langue sélectionnée

- CTc5sc05: Rendre un programme exempt d'erreurs

➤ **Composant 6 (CTc6): Démarche itérative**

- CTc6sc01: Mettre à l'essai (tester) d'un programme ou solution et analyser des erreurs éventuelles

○ CTc6sc02: S'approcher d'une solution par une démarche d'essai-erreur

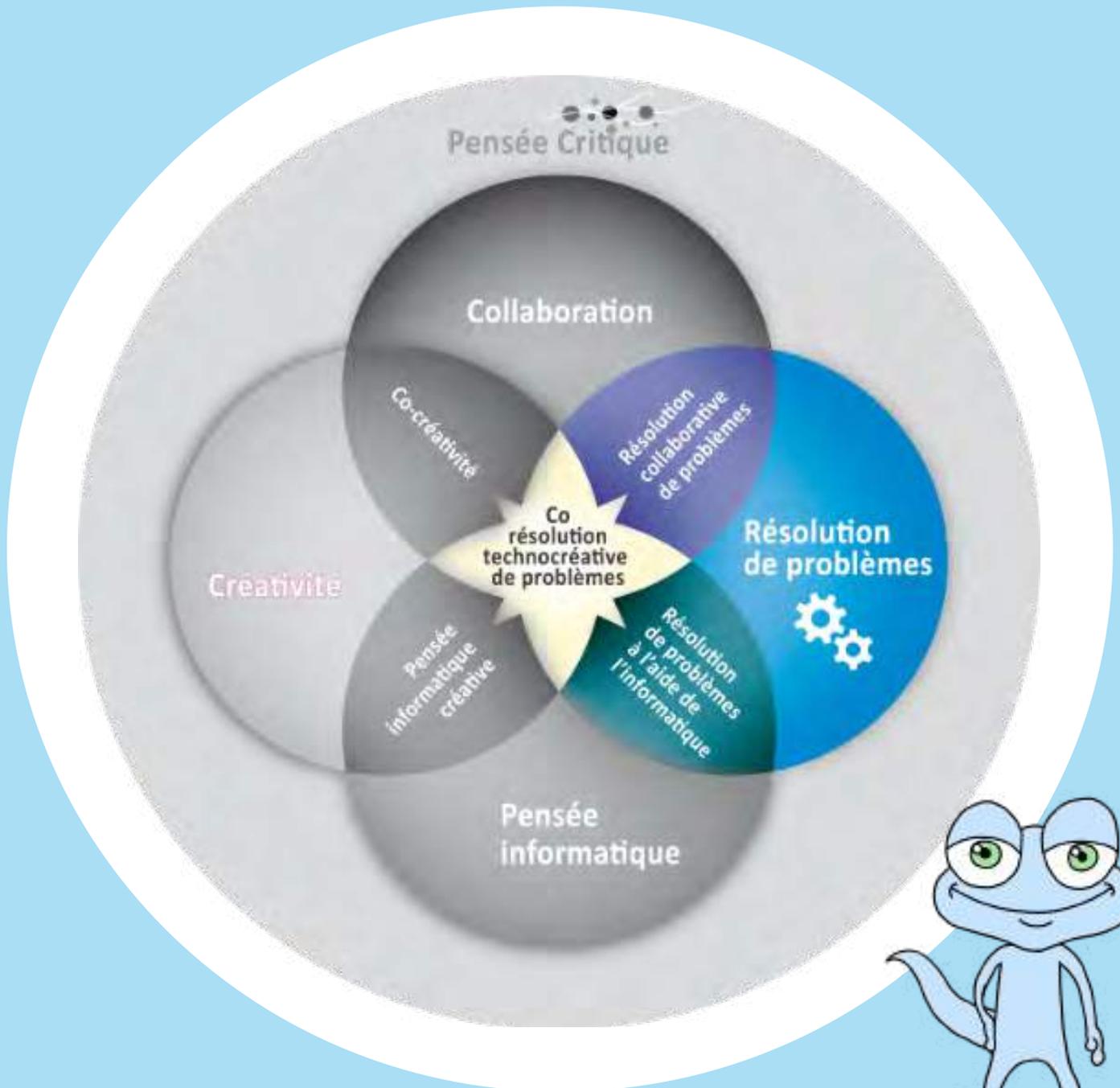
- CTc6sc03: Adopter une démarche itérative basée sur l'optimisation du

programme

- CTc6sc04: Tenir compte de l'état (initial, en cours ou final) du programme

- CTc6sc05: Évaluer la pertinence des objets représentés/opérationnaliser

dans un programme par rapport à une situation-problème



La **résolution de problèmes** est la capacité d'identifier une situation-problème pour laquelle le processus et la solution ne sont pas connus d'avance. C'est également la capacité de déterminer une solution, de la construire et de la mettre en œuvre de manière efficace.

➤ **Composant 1 (RP-a1): Établir et maintenir une compréhension partagée**

- RP-a1: Découvrir les points de vue et les habiletés des autres membres de l'équipe
- RP-b1: Construire une représentation partagée et négocier le sens du problème ou de l'activité à réaliser (espace partagé)
- RP-c1: Communiquer avec les membres de l'équipe à propos des actions à entreprendre ou étant entreprises
- RP-d1: Monitorer » et réaligner la compréhension partagée

➤ **Composant 2 (RP-a2): Entreprendre des actions appropriées pour résoudre le problème**

- RP-a2: Découvrir le type d'interaction collaborative pour résoudre le problème ainsi que les buts
- RP-b2: (ind) Identifier et décrire les tâches à compléter
- RP-c2: (ind) Mettre les plans en oeuvre
- RP-d2: « Monitorer » les résultats des actions et évaluer le succès de la résolution de problèmes (D2-ind) Suivre le plan et réguler sa propre activité.

➤ **Composant 3 (RP-a3): Établir et maintenir l'organisation de l'équipe**

- RP-a3: Comprendre les rôles pour résoudre un problème
- RP-b3: Décrire les rôles et l'organisation de l'équipe (protocole de communication/règles d'engagement)
- RP-c3: Suivre les règles d'engagement (par exemple, inciter les autres membres de l'équipe à accomplir leurs tâches.)
- RP-d3: « Monitorer », fournir des rétroactions et adapter l'organisation de l'équipe et les rôles.

➤ **Composant 4 (RP-a4): Co-régulation itérative des solutions intermédiaires**

- RP-a4: (ind) Développer la confiance, l'acceptation de l'échec et la résilience lors de l'évaluation des solutions intermédiaires au problème.
- RP-b4: (ind) Être capable de concevoir la résolution de problèmes comme une approche itérative orientée vers des prototypes
- RP-c4: Co-réguler des efforts comme itérations de solutions intermédiaires qui conduisent vers une solution optimale
(C4-ind) Réguler des efforts comme itérations de solutions intermédiaires qui conduisent vers une solution optimale
- RP-d4: (ind) Évaluer les avantages et inconvénients des solutions intermédiaires et adapter des prochaines itérations en conséquence

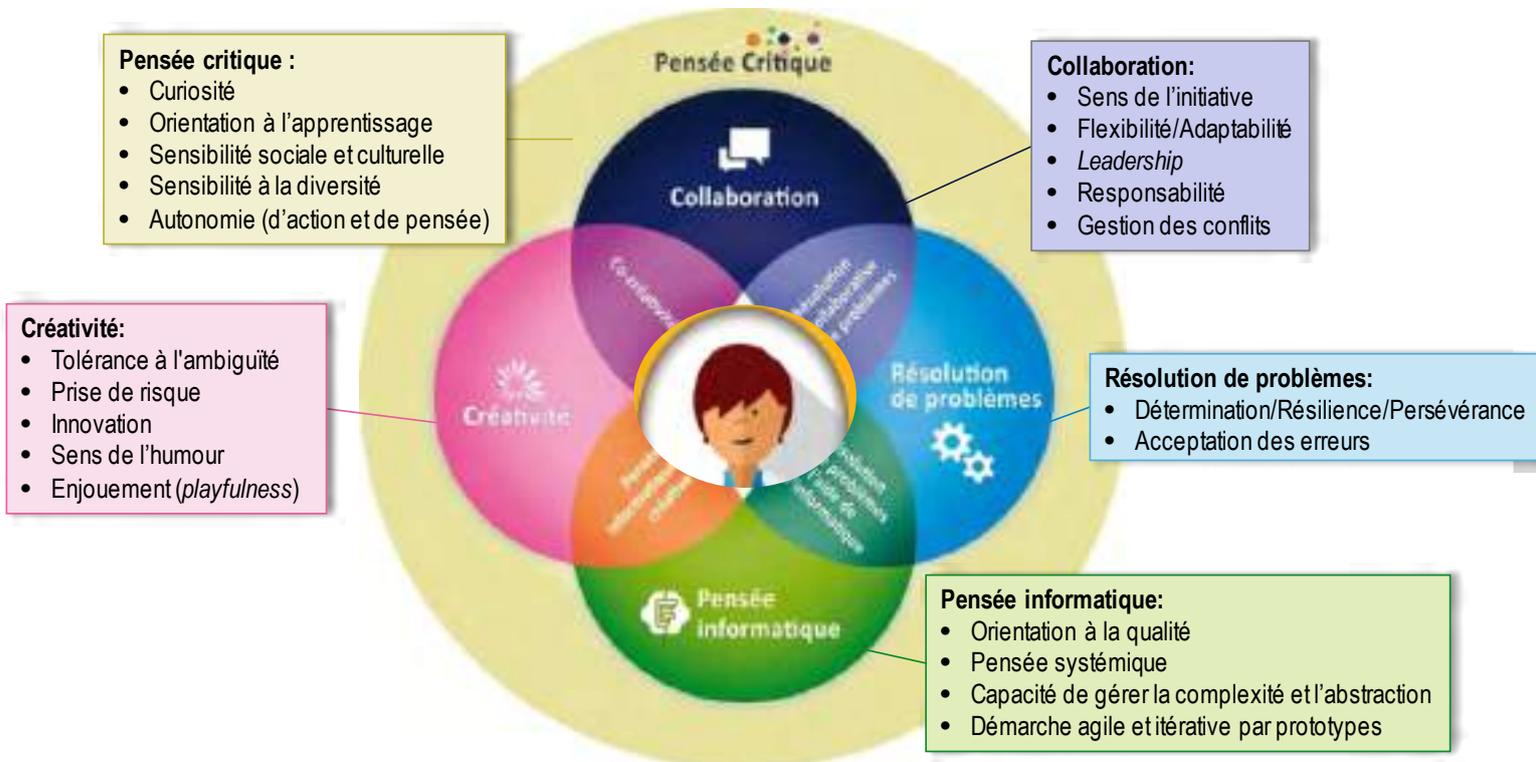
➤ **Composant 5 (RP-a5): Recherche et partage des ressources externes**

- RP-a5: Rechercher les ressources externes qui peuvent aider à résoudre le problème
- RP-b5: Extraire et partager les informations avec les autres membres de l'équipe
- RP-c5: Se mettre d'accord avec les autres membres de l'équipe sur la façon d'utiliser les ressources
- RP-d5: Évaluer avec les autres membres de l'équipe l'efficacité des ressources utilisées (D5-ind) Évaluer l'efficacité des ressources utilisées



Pistes pour l'évaluation des défis selon les Attitudes individuelles en lien aux compétences du 21e siècle (#5c21)

Attitudes individuelles en lien aux compétences 21^e siècle #5c21



Pistes pour l'évaluation des défis selon les Valeurs pour la communauté d'apprentissage (#5c21)



Valeurs pour la communauté d'apprentissage #5c21

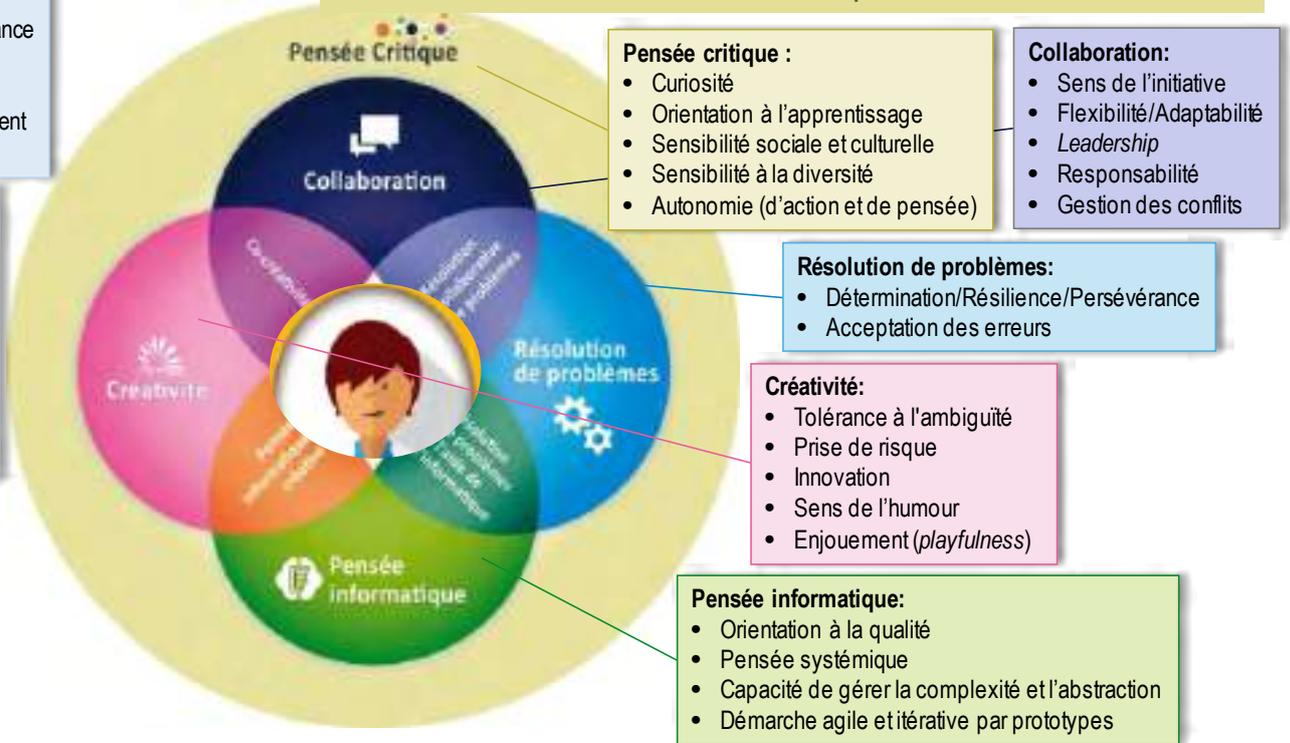
Groupe:

- Entraide
- Équité
- Confiance
- Interdépendance
- Courtoisie et calme
- Développement durable

Apprenant:

- Hédonisme
- Réussite
- Qualité
- Liberté
- Intégrité
- Privacité
- Usabilité
- Autonomie

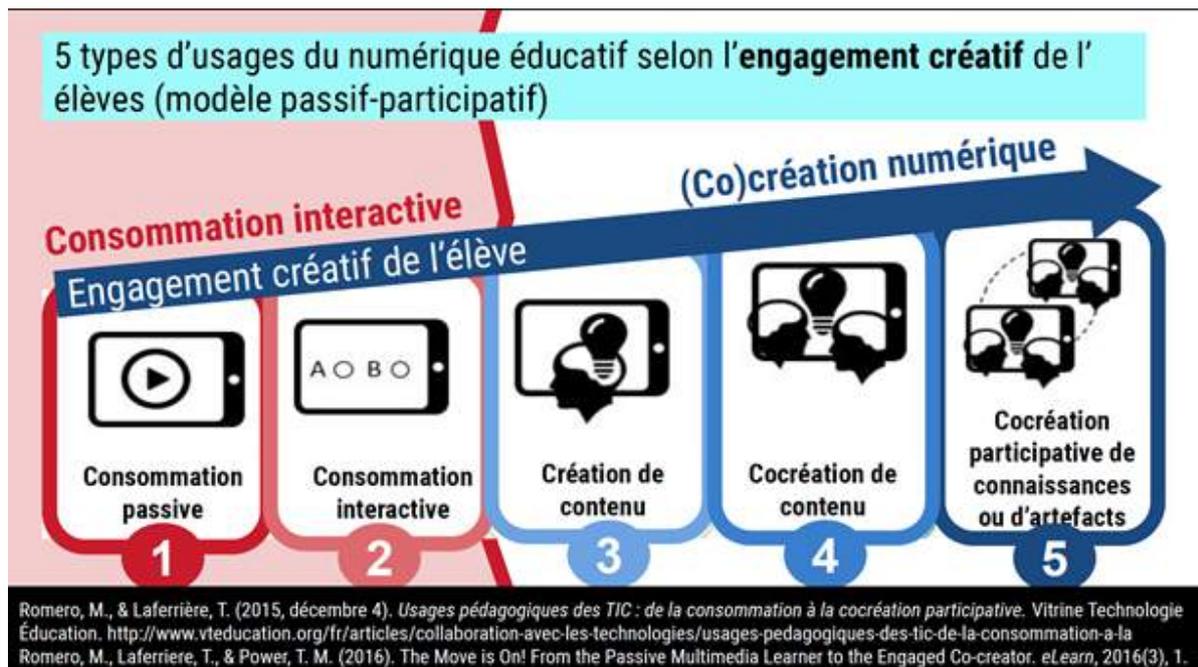
Attitudes individuelles en lien aux compétences 21^e siècle #5c21





Pistes pour l'évaluation des défis selon le Engagement créatif (#5c21)

L'engagement créatif peut être utilisé numériquement ou débranché, l'élève pouvant effectuer ses activités de manière kinesthésique. Voir plus de conseils dans l'article cité ci-dessous.



D'autres ressources

Canvas

<https://scratch.mit.edu/projects/119727640/#editor>

In progress:

<https://scratch.mit.edu/projects/128326212/#editor>

<https://scratch.mit.edu/projects/123879338/#editor>

BIBLIOGRAPHIE

Adapté du travail: Romero, M., Roy, A, Lepage, A. (2016). Défis de programmation créative: du conte au code avec Scratch et Vibot. Université Laval-Québec. Disponible em: <https://goo.gl/bBWGIk> (Publicação original).

Romero, M. et Vallerand, V. (2016). Guide d'activités technocréatives pour les enfants du 21e siècle. Université Laval-Québec. Disponible em: <https://lel.crires.ulaval.ca/oeuvre/guide-dactivites-technocreatives-pour-les-enfants-du-21e-siecle>

Romero, M.; Vallerand, V.; Nunes, M. A. S. N. (2019) Almanaque Para Popularização De Ciência Da Computação. Série 12: Guia Pedagógico; Volume 1: Atividades Tecnocriativas para crianças do século 21. ed. 1. Porto Alegre: SBC. v. 1. Disponible em: <http://almanaquesdacomputacao.com.br/gutanunes/publications/S12V1.pdf>

Romero, M.; Insfrán, D. A.; Nunes, M. A. S. N.; Loufane, S.F. (2019) Almanaque Para Popularização De Ciência Da Computação. Série 12: Guia Pedagógico; Volume 5: Betabot - a representante brasileira da Liga dos Bots para o desenvolvimento do pensamento computacional no Brasil. ed. 1. Porto Alegre: SBC. v. 5. Disponible em: <http://almanaquesdacomputacao.com.br/gutanunes/publications/S12V5.pdf>

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. International Journal of Educational Technology in Higher Education, 14(1), 42.

Romero, M. et al. (2017). Assessment of creativity and computational thinking in Scratch projects. In: Scratch conference. #Scratch2017BDX. Disponible em: <https://vibot.fse.ulaval.ca/2017/07/19/scratch2017bdx-assessment-of-creativity-and-computational-thinking-in-scratch-projects/>

Romero, M. (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative. Formation et profession, 24(1), 87-89. <https://doi.org/10.18162/fp.2016.a92>

Romero, M., Laferriere, T., & Power, T. M. (2016). The Move is On! From the Passive Multimedia Learner to the Engaged Co-creator. eLearn, 2016(3), 1.
Romero, M., Davidson, A-L., Cucinelli, G., Ouellet, H., & Arthur, K. (2016). Learning to code: from procedural puzzle-based games to creative programming. CIDUI.

Romero, M., & Laferrière, T. (2015, décembre 4). Usages pédagogiques des TIC : de la consommation à la cocréation participative. Vitrine Technologie Éducation. <http://www.vteducation.org/fr/articles/collaboration-avec-les-technologies/usages-pedagogiques-des-tic-de-la-consommation-a-la>

Romero, M., Laferriere, T., & Power, T. M. (2016). The Move is On! From the Passive Multimedia Learner to the Engaged Co-creator. eLearn, 2016(3), 1.
Romero, M. e Loufane, S.F. Vibot the Robot: Introduction to programming. Université Laval/ Québec. Ed. Publications Québec. p.24. 2016. Disponível em <https://www.amazon.fr/>

Yağcı, M. A valid and reliable tool for examining computational thinking skills. IN: Educ Inf Technol (2019). 24: 929. <https://doi.org/10.1007/s10639-018-9801-8>

D'autres bandes dessinées, brochures et guides sur:
<http://almanaguesdacomputacao.com.br/>

AUTORES

Margarida Romero



Elle est directrice de recherche au Laboratoire d'Innovation et Numériques pour l'Education (LINE), un laboratoire d'apprentissage assisté par la technologie (TEL). Elle est professeure titulaire à l'Université Cotê d'Azur (France) et professeure associée à l'Université Laval de (Canada). Ses recherches portent sur les utilisations inclusives, humanistes et créatives des technologies (co-conception, conception de jeux et robotique) pour

le développement de la créativité, la résolution de problèmes, la collaboration et la pensée informatique. Elle est responsable de la conception philosophique, de la planification et de la création de la version conceptuelle de Vibot

<https://margaridaromero.wordpress.com/>

Linkedin: <https://www.linkedin.com/in/margarida/>

Maria Augusta Silveira Netto Nunes

Bolsista de Produtividade Desen. Tec. e Extensão Inovadora do CNPq - Nível 2 - CA 96 - Programa de Desenvolvimento Tecnológico e Industrial



Elle est professeure associé II du département d'informatique de l'université Fédérale de Sergipe. Membre du programme de Mastere en

informatique (PROCC) à UFS. Post-doctorat au laboratoire LINE, Université de la Côte d'Azur / Nice Sophia Antipolis / Nice-France (2019). Post-doctorat de l'Institut national de la propriété industrielle (INPI) (2016). Doctorat en informatique de l'Université de Montpellier II - LIRMM à Montpellier, France (2008). Stage de doctorat (doc-sandwich) à l'INESC-ID-IST Lisbonne-Portugal (août 2007-février 2008). Master en sciences informatiques à Université fédérale de Rio Grande do Sul (1998) Diplômée en informatique de

l'Université de Passo Fundo-RS (1995). Elle est chercheuse en productivité DT-CNPq Actuellement, ses recherches portent principalement sur la pensée informtaique à l'aide de bandes dessinées. Création du projet "Almanagues pour la vulgarisation de l'informatique" parrainé par SBC.

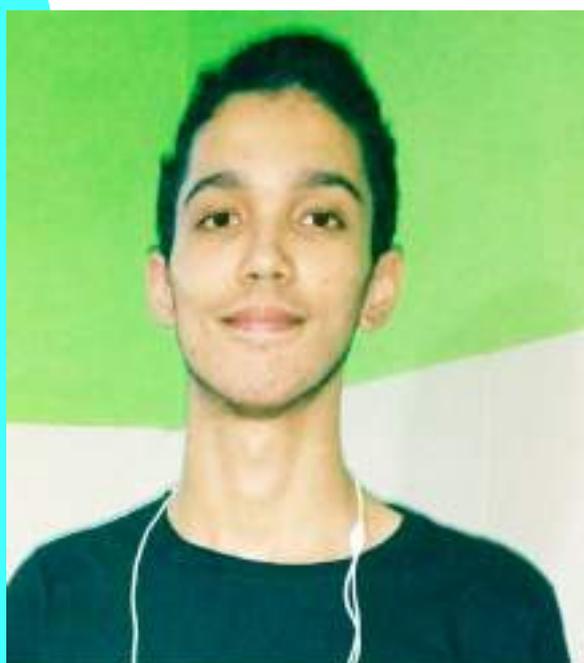
Lattes: <http://lattes.cnpq.br/9923270028346687>



Luís Antônio dos Santos Silva

Il est un technicien diplômé de l'Institut fédéral de Sergipe (IFS). Diplômé du Bachelor en systèmes d'information de l'Université Tiradentes (UNIT), actuellement étudiant du programme d'études supérieures en informatique (PROCC) de l'Université fédérale de Sergipe (UFS). Actuellement, ses recherches sont principalement axées sur la pensée informatique.

Lattes: <http://lattes.cnpq.br/1622211414785978>



José Humberto dos Santos Júnior Bolsista CNPq - Iniciação Tecnológica

Étudiant en informatique à l'Université fédérale de Sergipe - UFS.

Lattes: <http://lattes.cnpq.br/9144803555676838>



Aurélie Roy

Elle est étudiante la littérature anglaise à l'Université Laval-Québec / CA. Son fort intérêt pour les langues, l'éducation et le design lui a permis de rejoindre l'équipe de #CoCreaTIC en 2016. Elle a récemment développé une passion pour la programmation créative. Elle travaille avec Scratch. Bientôt commence son doctorat en littérature anglaise



Alexandre Lepage

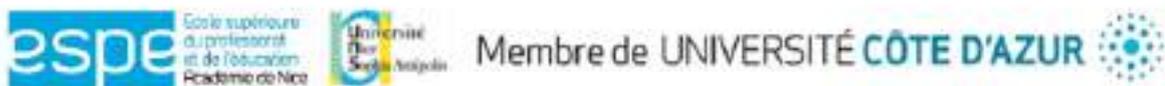
Il est enseignant au lycée. Il poursuit une maîtrise en technologie de l'éducation à l'Université Laval-Québec / CA. Intéressé par l'apprentissage de la programmation pour développer des stratégies métacognitives et les appliquer à l'enseignement.

LinkedIn: <https://www.linkedin.com/in/alexandre-lepage-8b395351/>

Remerciements spéciaux

Ao CNPq, CAPES, SBC, BICEN, DCOMP/PROCC, UNIRIO/UNIRIOTEC, Université Laval, Publications Québec, Université Nice Sophia Antipolis, Université Côte d'Azur, LINE- Laboratoire.

SOUTIEN:



ISBN 978-857669492-2



9

788576

694922