

Module 1 - partie 2 chapitre 2 Débugger

Fiche d'identité

Résumé de l'activité	Débugger des programmes, et créer son propre programme buggé !
Objectif(s)	<ul style="list-style-type: none"> Savoir analyser un programme et trouver les problèmes Trouver une solution adaptée à chaque programme

Durée approximative de l'activité	<ul style="list-style-type: none"> 1h30
Participants	<ul style="list-style-type: none"> jusqu'à 15 enfants à partir de 8 ans
Matériel nécessaire	<ul style="list-style-type: none"> Au moins 1 ordinateur pour 2 enfants, connecté à Internet Un vidéo projecteur si possible
Préparation	<ul style="list-style-type: none"> Aucune

Notions liées	<i>programmation, analyser un programme, debugger</i>
Lien éventuel avec le programme scolaire	

Déroulement

1 Donner la consigne – 5'	
Aujourd'hui, nous allons partir à la chasse aux bugs !	<ul style="list-style-type: none"> <i>Expliquer ce qu'est un bug, et d'où ça vient !</i>

2 Débugger des programmes existants – 25'	
On va commencer par debugger des programmes déjà existants	<ul style="list-style-type: none"> <i>Vous pouvez utiliser le studio suivant : https://scratch.mit.edu/studios/188628/ (et éventuellement celui ci https://scratch.mit.edu/studios/475539/ ou celui là : https://scratch.mit.edu/studios/475554/)</i>
Ce projet ne marche pas, flûte !	<ul style="list-style-type: none"> <i>Avec les enfants, montrez leur comment vous faites pour trouver et résoudre un bug : exécuter le programme pour voir ce qui ne va pas, lire le programme pour comprendre</i>

< Class'Code >

		son fonctionnement et trouver la partie qui ne marche pas et la remplacer !
	À vous de jouer !	<ul style="list-style-type: none"> ▪ Installer les enfants de manière à ce qu'ils soient en situation d'écoute et d'échange ▪ Chacun regarde les projets et essaie de résoudre par lui même. On évite de donner la réponse au voisin trop rapidement, il faut qu'il cherche lui aussi ;)
	Est-ce que vous avez trouvé tous les bugs ?	<ul style="list-style-type: none"> ▪ Faites un petit tour des projets que vous leur avez demandé de debugger, et s'il y en a qui posent problème voir avec eux comment les résoudre.

3	Créer des projets buggés - 60'	
	À notre tour de créer des projets buggés !	<ul style="list-style-type: none"> ▪ Chacun crée un (ou deux) projets buggés que les autres devront résoudre. ▪ Pensez à bien expliquer dans les "instructions" quel est le problème du projet s'il n'est pas évident
	On partage tout cela.	<ul style="list-style-type: none"> ▪ Créez un studio pour que chaque enfant puisse ajouter ses projets buggés au fur et à mesure !
	Pas besoin de passer trop de temps sur les graphismes de votre projet	<ul style="list-style-type: none"> ▪ L'important est de voir le code du projet
	Maintenant qu'on a suffisamment de projets buggés, on voudrait les résoudre !	<ul style="list-style-type: none"> ▪ Le fonctionnement est le même que pour l'étape 2

5	Conclusion - 5'	
	Pourquoi doit on debugger ?	<ul style="list-style-type: none"> ▪ Savoir debugger soit-même permet d'être autonome pour résoudre ses problèmes plutôt que de devoir attendre que quelqu'un vienne vous aider ▪ Tout le monde fait des erreurs et laisse des bugs dans son code ! C'est normal d'en rencontrer ▪ Savoir lire et comprendre le script de quelqu'un c'est pratique quand on trouve un bout de code sur Scratch et qu'on ne sait pas ce qu'il fait.
	<ul style="list-style-type: none"> ▪ Qu'est-ce que j'ai appris ? ▪ Qu'est-ce qui était difficile ? 	<ul style="list-style-type: none"> ▪ Permettre aux enfants d'exprimer ce qu'ils ont pensé de l'atelier, de poser des questions, etc. ▪ Leur expliquer ce qu'ils vont faire pendant la séance suivante, le cas

<Class'Code>

échéant :)

Pour aller plus loin

Dis papa (ou maman), comment arrivent les bugs dans le monde numérique ? par Aurélien Alvarez et Thierry Viéville, Images des mathématiques, 2014

<http://images.math.cnrs.fr/Dis-papa-ou-maman-comment-arrivent-les-bugs-dans-le-monde-numerique>