

## 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"> <li>Savoir analyser un programme et trouver les problèmes</li> <li>Trouver une solution adaptée à chaque programme</li> </ul>

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

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

Déroulement

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

<b>2 Debugger des programmes existants – 25'</b>	
On va commencer par debugger des programmes déjà existants	<ul style="list-style-type: none"> <li><i>Vous pouvez utiliser le studio suivant : <a href="https://scratch.mit.edu/studios/188628/">https://scratch.mit.edu/studios/188628/</a> (et éventuellement celui ci <a href="https://scratch.mit.edu/studios/475539/">https://scratch.mit.edu/studios/475539/</a> ou celui là : <a href="https://scratch.mit.edu/studios/475554/">https://scratch.mit.edu/studios/475554/</a>)</i></li> </ul>
Ce projet ne marche pas, flûte !	<ul style="list-style-type: none"> <li><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></li> </ul>

# < Class'Code >

		<i>son fonctionnement et trouver la partie qui ne marche pas et la remplacer !</i>
	À vous de jouer !	<ul style="list-style-type: none"><li>▪ <i>Installer les enfants de manière à ce qu'ils soient en situation d'écoute et d'échange</i></li><li>▪ <i>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 ;)</i></li></ul>
	Est-ce que vous avez trouvé tous les bugs ?	<ul style="list-style-type: none"><li>▪ <i>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.</i></li></ul>

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

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

# <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>