

Codage à l'école : Snap! ou Scratch : il y a urgence

-----  
(.. ..)  
( - )  
( --- )

Bonjour,

Le codage arrive partout.

Obama a même fait une vidéo dans laquelle il engage les jeunes américains à apprendre à coder.

"President Obama asks America to learn computer science" \_(1)

En France, dans l'Education Nationale, il y a déjà quelques années que les algorithmes sont entrés dans les programmes de lycée (une vingtaine d'années déjà...). Mais la programmation en tant que telle n'est entrée dans les programmes de lycée que très récemment (Rentrée 2012) avec la spécialité ISN en Terminale S.

Quelques enseignants seulement allaient jusqu'à la phase de codage lorsqu'ils élaboraient des algorithmes avec les élèves en classe de Mathématiques, soit avec un logiciel, soit avec une calculatrice.

A cette rentrée, le codage va être officiellement dans les programmes de collège et d'école primaire. Je dis bien codage, pas seulement algorithmique. Mais il va falloir être extrêmement vigilants, car il serait question que le Ministère de l'Education Nationale souhaite imposer le logiciel Scratch \_(2) à tous. Ce serait dramatique en terme de libertés logicielle. Je vous explique.

En effet, Scratch est un logiciel qui sort des labos du MIT (Massachusetts Institute of Technology, grande Université du Massachusetts). Scratch est un logiciel libre (voir license), jusqu'à la version 1.4, j'étais emballée car on pouvait produire du code sans écrire une seule ligne. Scratch est un logiciel de programmation visuelle, qui marche avec des briques de code à emboîter les unes aux autres (comme des légos) pour produire son code. Très répandu aux Etats-Unis et au Canada, j'ai même vu des Haïtiens utiliser Scratch en classe en créole Haïtien. Je l'utilisais donc avec les élèves en classe de seconde pour transcrire des algorithmes de mathématiques.

Et cela plaisait aux élèves. Des enfants peuvent coder en Scratch. C'est ludique, rigolo, et amène tranquillement à faire de la programmation parallèle.

Jusque là tout est rose. Mais, à partir de la version 2, mise à jour majeure de Scratch, l'installation de ce logiciel nécessite l'installation d'Adobe Air, logiciel propriétaire par excellence...

Scratch 2 étant toujours un logiciel libre, l'université de Berkeley en a profité pour développer Snap!

\_(3), un logiciel de programmation fonctionnelle \_(4) visuelle à partir du code de Scratch. Basé sur le même principe, on déplace des briques de code, puis on les imbrique de façon à élaborer son programme petit à petit.

Snap! est entièrement libre, potentiellement beaucoup plus puissant que Scratch puisque la programmation fonctionnelle part du principe que tout

calcul est l'évaluation de fonctions mathématiques. Mais surtout, l'apport majeur que Snap! apporte à Scratch, c'est que l'on peut créer ses propres blocs.

En clair, Snap! permet de Tout faire. Je suis certaine que même les développeurs y trouveront leur compte :

en effet, Snap! est entièrement codé en HTML5+Javascript et on peut insérer du code Javascript en dur dans un code Snap! .

Puisque Snap! est codé en HTML5, son exécutable est une page HTML \_(5). Mais, si vous enregistrez la page Web complète (ou mieux encore, téléchargez les sources), vous trouverez dans les fichiers liés à la page principale tous les fichiers javascript (.js) qui construisent le logiciel. Le tout ne pèse que 1,4 Mo, et MARCHE HORS CONNECTION. Dans mon lycée où il est quasi-impossible de faire surfer plusieurs classes en même temps, c'est essentiel...

On peut ajouter à cela les quelques outils et bibliothèques qui peuvent s'ouvrir à la demande. Ils se téléchargent et sont au format XML. Donc encore seulement quelques kiloOctets.

Les projets Snap! se sauvegardent au format XML (toujours seulement quelques kiloOctets), et un logiciel open-source que l'on trouve en ligne \_(6) vous permet si vous le souhaitez de créer un exécutable "stand-alone" de votre projet, et cela est proposé pour les 3 plate-formes : Linux, Mac et Windows !

Inimaginable... L'ouverture de l'équipe qui tourne autour de Snap! me stupéfie. Ce sont des gens géniaux qui sont derrière.

Une page HTML est dédiée à la transcription de projets Scratch en Snap. \_(7)

Hier, j'ai ouvert un thread sur le Github du développeur (Jens Mönig) qui s'intitulait "Blocks with sprites". Ce développeur est ouvert, réactif et répond à toutes les questions !!! Il a même pris le temps de faire des vidéos pour que je comprenne mieux !!!

Lors de ces échanges, il a aussi expliqué que l'implémentation du passage en argument de sprites (des objets visuels) dans des blocs sera effective à la prochaine mise à jour majeure : la version 4.1 (nous en sommes à la 4.0.5).

Qui plus est, ce développeur semble avoir des bases théoriques de haut niveau puisque la mascotte de Snap! s'appelle Alonzo (en hommage à Alonzo Church \_(8), le père du lambda-calcul \_(9), et donc, indirectement, qui impactera sur la programmation fonctionnelle).

En plus, Alonzo a une mèche de cheveu en forme de lambda !

En conclusion, il faut rester très vigilant, et devancer le problème, car, du jour au lendemain, un Bulletin Officiel (B.O.) peut sortir du Ministère de l'Education Nationale, qui obligerait les enseignants à utiliser Scratch à la rentrée prochaine. Le lobbie derrière tout ceci, c'est que les rectorats pourront distribuer à terme des tablettes Windows dans tous les établissements... L'accord avec Microsoft ayant déjà été signé.

Il NE FAUT PAS que le logiciel Scratch soit nominativement inscrit dans les programmes officiels du B.O. qui paraîtra très bientôt pour la prochaine rentrée.

J'informe l'April en parallèle.

Allez, n'hésitez pas à regarder du côté de Snap!

Nathalie Carrié, enseignante en Mathématiques au lycée.

- \_(1) <https://www.youtube.com/watch?v=6XvmhE1J9PY>
- \_(2) <https://scratch.mit.edu/>
- \_(3) <http://snap.berkeley.edu/>
- \_(4) [https://fr.wikipedia.org/wiki/Programmation\\_fonctionnelle](https://fr.wikipedia.org/wiki/Programmation_fonctionnelle)
- \_(5) <http://snap.berkeley.edu/snapsource/snap.html>
- \_(6) <http://snapp.citilab.eu/>
- \_(7) <http://djdolphin.github.io/Snapin8r2/>
- \_(8) [https://fr.wikipedia.org/wiki/Alonzo\\_Church](https://fr.wikipedia.org/wiki/Alonzo_Church)
- \_(9) <https://fr.wikipedia.org/wiki/Lambda-calcul>