

Qu' **apprennent** les étudiants quand ils créent des histoires interactives, des animations, des jeux, de la musique et une activité artistique avec Scratch ?

D'une part, ils **apprennent les idées mathématiques et logico-déductives** qui se trouvent intégrées dans Scratch. Quand les étudiants créent des programmes dans Scratch, ils apprennent des concepts essentiels d'algorithmique de base tels que les processus itératifs (les boucles par exemple) et les instructions conditionnelles (si, si-sinon). Ils acquièrent également une compréhension de concepts mathématiques importants tels que les coordonnées (dans un repère du plan) les variables et les nombres aléatoires.

D'autre part, les étudiants apprennent ces concepts dans un environnement **significatif et motivant**. Quand les étudiants apprennent des variables dans un contexte traditionnel d'enseignement de d'algèbre, ils se sentent assez peu concerné par le concept quelque peu abstrait. Mais quand ils apprennent des variables avec Scratch ils peuvent les utiliser immédiatement d'une façon très concrète : pour contrôler la vitesse d'une animation, ou garder la trace du score dans un jeu qu'ils sont en train de créer.

Quand les étudiants travaillent sur les projets Scratch, ils mettent en oeuvre **le processus de conception** d'un projet. Typiquement, un étudiant va commencer avec une idée. Il crée un prototype fonctionnel, apporte et expérimente lui-même ses solutions, le corrige quand les

choses fonctionnent pas comme prévu. Par suite il tente d'obtenir l'avis de ses pairs, puis il met à jour selon les retours obtenus et enfin il remodélise. C'est une spirale continue : avoir une idée, créer un projet, qui amène de nouvelles idées, qui inspire de nouveaux projets et ainsi de suite.

Ce processus de conception d'un projet sollicite et développe de nombreuses **compétences nécessaires aux jeunes du 21^e siècle** qui désirent avoir du succès plus tard dans leur vie : émettre des idées créatives, communiquer clairement, analyser systématiquement, collaborer efficacement avec les partenaires, concevoir itérativement, et se former en permanence.

La création des projets dans Scratch aide aussi les étudiants à atteindre un niveau plus approfondi de **maîtrise (fluency)*** avec la technologie numérique. Que voulons-nous dire par la **maîtrise (fluency)*** ? Pour être considéré comme à l'aise (*fluent*)* en anglais, espagnol, ou d'autre langue, il ne suffit pas seulement d'apprendre à lire, mais aussi à écrire ; c'est à dire savoir comment s'exprimer avec cette langue. De même, pour être à l'aise (*fluent*)* avec la technologie numérique, vous ne devez pas seulement apprendre comment utiliser, interagir avec l'ordinateur, mais aussi apprendre à créer avec lui.

Bien entendu, la plupart des étudiants ne deviendront pas tous des programmeurs professionnels, tout comme la plupart d'entre-eux ne deviendront pas des auteurs professionnels. Mais **apprendre à programmer** est bénéfique pour chacun d'entre nous : il permet aux étudiants de s'exprimer pleinement de façon créative, les aide à se développer leur sens logique et à comprendre le fonctionnement des nouvelles technologies qu'ils rencontrent partout dans leur vie quotidienne.

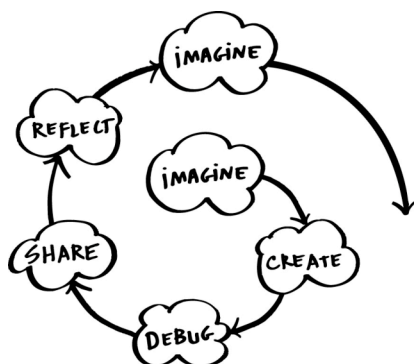
Références

(*) Rethinking Learning in the Digital Age

<http://www.media.mit.edu/~mres/papers/wef.pdf>

(*) Learning for the 21st Century

(<http://www.21stcenturyskills.org/>)



Lifelong Kindergarten Group, MIT Media Lab