Human Resource Machine (descriptif d'atelier)

Christian Blanvillain

Human Resource Machine

Christian Blanvillain

HEP Vaud — UER Média et TIC
christian.blanvillain@hepl.ch

Résumé

La société Tomorrow Corporation a développé un jeu de programmation visuelle Human resource machine, proche de l'assembleur, disponible sur quasiment toutes les plateformes. Nous avons repris leur métaphore pour faire redécouvrir des concepts de base de la programmation aux élèves à travers une activité jouée en classe. C'est cette activité, sous une forme de démarche auto-socio-constructiviste, qui est proposée dans la première partie de cet atelier.
À l'issue de l'expérience récréative, une analyse réflexive est proposée dans la seconde partie de l'atelier. Nous introduisons alors le schéma du fonctionnement cognitif qui représente l'intelligence comme une organisation de fonctions cognitives, ouvrant ainsi la porte à une ludification des mécanismes métacognitifs. Pour l'enseignant, il ne s'agit plus seulement de faire apprendre des savoirs, mais de faire conscientiser par l'apprenant la nécessaire maîtrise des processus élémentaires qui sous-tendent tout apprentissage.
Ce nouveau rôle pour l'enseignant vient s'ajouter à celui de pédagogue et de didacticien. Nous conclurons cet atelier par la présentation d'une version enrichie du triangle de Houssaye, où sera mise en évidence la nécessité d'une science émergente : la noosologie, qui se préoccupe d'enseigner comment apprendre à apprendre.

Mots clés : serious game, programmation, auto-socio-constructivisme, métacognition, noosologie

1 Manipulations

Une équipe de 2 à 7 personnes doivent résoudre un problème algorithmique de programmation à l'aide d'un micro langage proche de l'assembleur, similaire au jeu Human resource machine (Figure 1) (Blomquist, Gabler & Gray, 2015) développé par la société Tomorrow corporation (Blomquist, Gabler & Gray, 2010). Chacun a un rôle bien défini. Les programmes sont élaborés à l'aide de post-it magnétiques sur un tableau blanc. Des fiches de problèmes de différents niveaux reprennent les challenges du jeu et en proposent de nouveaux.
Deux équipes sont mises en concurrence de manière à limiter le temps de recherche. À l'issue de l'activité, il est demandé aux équipes d'analyser le code de l'autre équipe. Le constat qui devrait émerger du groupe, c'est que l'assembleur est difficile à relire. L'animateur peut alors leur proposer de mieux structurer leur code, ce qui est le moyen d'introduire les procédures, les boucles, les structures de codes et d'abolir les instructions jump.
Dans le contexte du colloque, pour expliquer l'environnement nous présenterons un niveau d'initiation dit « d'égalisation » (niveau 13 dans le jeu original, durée 5mn) adapté aux connaissances du public informaticien. Lorsque les règles du jeu sont comprises, nous proposons le challenge de « maximisation » (niveau 14 dans le jeu original, durée 5mn) consistant à identifier le plus grand de deux nombres. Avec le jeu d'instructions disponibles, il est impossible de résoudre ce challenge avec seulement 10 lignes de code. Le groupe devra alors inventer une instruction supplémentaire, le saut conditionnel négatif, pour pouvoir l'utiliser. Il y a moyen de le résoudre sans, mais l'animateur devra alors poser la question de comment faire un code plus court et plus élégant pour susciter la démarche de réflexion du groupe.
Cette démarche auto-socio-constructiviste (Bassis, 2011), dans la suite des idées constructivistes de Piaget et socioconstructivistes de Wallon, Vygotski et Bruner, peut être reproduite sans difficulté en classe par tout enseignant qui cherche à initier les élèves à l'algorithmie et la programmation.

2 Analyse réflexive

Une fois le challenge terminé, nous lancerons les participants sur une analyse réflexive des processus cognitifs mis en œuvre pour résoudre l'énigme proposée et sur les transferts possibles pour un usage dans leurs classes. L'intention étant double : apprendre à programmer et expliciter les mécanismes cognitifs utilisés pour programmer. Après 5mn de débats, nous présentons le schéma du fonctionnement cognitif selon Vianin (2009) (Figure 2), qui représente les mécanismes de l'intelligence comme une organisation de fonctions cognitives extrêmement semblables au gameplay du jeu de programmation et donc au fonctionnement d'un processeur. On y retrouve la prise d'information et l'expression de la réponse (input box et output box), le processeur central de traitement de l'information avec les processus cognitifs (tous les opérateurs arithmétiques) et métacognitifs (les sauts et instructions conditionnelles) qui déterminent le flux d'instructions et donc la stratégie caractérisée par la logique d'articulation des compétences élémentaires, et enfin la mémoire à moyen et long terme (copyFrom et copyTo).
Une fois ce parallélisme fait, nous terminerons la discussion en explorant l'importance en informatique d'étudier les mécanismes de la compréhension et de l'apprentissage chez l'élève. En effet, de la même manière que nous aurions du mal à expliciter tout ce que faisons pour pouvoir marcher, courir ou pédaler, avec nos années de pratique informatique, il est souvent très difficile d'identifier les difficultés que rencontrent nos élèves et ces dernières ne sont souvent pas liées aux problèmes que nous leur présentons, mais bel et bien aux automatismes qu'ils se doivent de construire. Le jeu *Human Resource Machine* est un excellent moyen pour entrer en discussion avec les élèves sur le sujet de leurs méthodes de pensée, tout en restant dans leur zone proximale de développement. Des évaluations diagnostiques des processus cognitifs mobilisés pour résoudre les énigmes du jeu peuvent aider les élèves à améliorer leurs méthodes d'apprentissage. Pierre Vianin, en partant des apports de Feuerstein, en propose quelques-unes directement réutilisables.
Mais ce n'est pas la seule source sur laquelle nous pouvons nous appuyer pour aider les élèves. Nous trouvons de formidables ressources auprès des neurosciences cognitives et de la psychologie cognitive. Des auteurs tels qu'André Giordan, Britt-Mari Barth, Antoine de La Garanderie, Howard Gardner et bien d'autres encore, apportent, chacun dans son domaine, un éclairage nouveau sur la question d'apprendre. Or dans la plupart des formations pédagogiques, ce rôle fondamental de l'enseignant n'est pas vraiment développé. Ainsi il conviendrait peut-être de combler ce manque et de regrouper sous un même nom ou plutôt une même science, ces méthodes et outils qui permettent d'aider l'autre à développer ses intelligences ? En référence à la conception philosophique tripartite de l'humain de l'antiquité grecque : psyché, sôma et noos (ou *noûs*), nous avons choisi d'utiliser la racine noos (l'esprit, l'intellect, la raison) pour désigner ce concept de science émergeante : la *noosologie*.
Nous terminons donc cet atelier par une revisite du triangle de Houssaye afin de compléter les deux dimensions pédagogiques et didactiques en ajoutant un troisième axe, celui d'apprendre à apprendre ou comment aider l'autre à développer ses intelligences (Figure 3).

3 Jeu d'instructions

inBox	Lire une valeur depuis le flux d'entrée
outBox	Déposer la valeur en cours sur le flux de sortie
copyFrom	Lire la valeur en cours depuis une case mémoire
copyTo	Ecrire la valeur en cours dans une case mémoire
add	Ajouter la valeur d'une case mémoire à la valeur en cours
sub	Soustraire la valeur d'une case mémoire à la valeur en cours
bump+	Augmenter la valeur entière d'une case mémoire
bump−	Diminuer la valeur entière d'une case mémoire
jumpTo	Sauter dans le code vers l'étiquette correspondante
jumpIfNull	Sauter dans le code vers l'étiquette correspondante si la valeur en cours est zéro

Références

Bassis, O. (2011). La démarche d'auto-socio-construction des savoirs. *Dialogue*, *120*.

Blomquist, A., Gabler, K. & Gray, K. (2010). *Tomorrow corporation.* (`http://tomorrowcorporation.com`)

Blomquist, A., Gabler, K. & Gray, K. (2015). *Human resource machine.* (`https://en.wikipedia.org/wiki/Human_Resource_Machine`)

Vianin, P. (2009). *L'aide stratégique aux élèves en difficulté scolaire – comment donner à l'élève les clés de sa réussite ?* De Boeck.

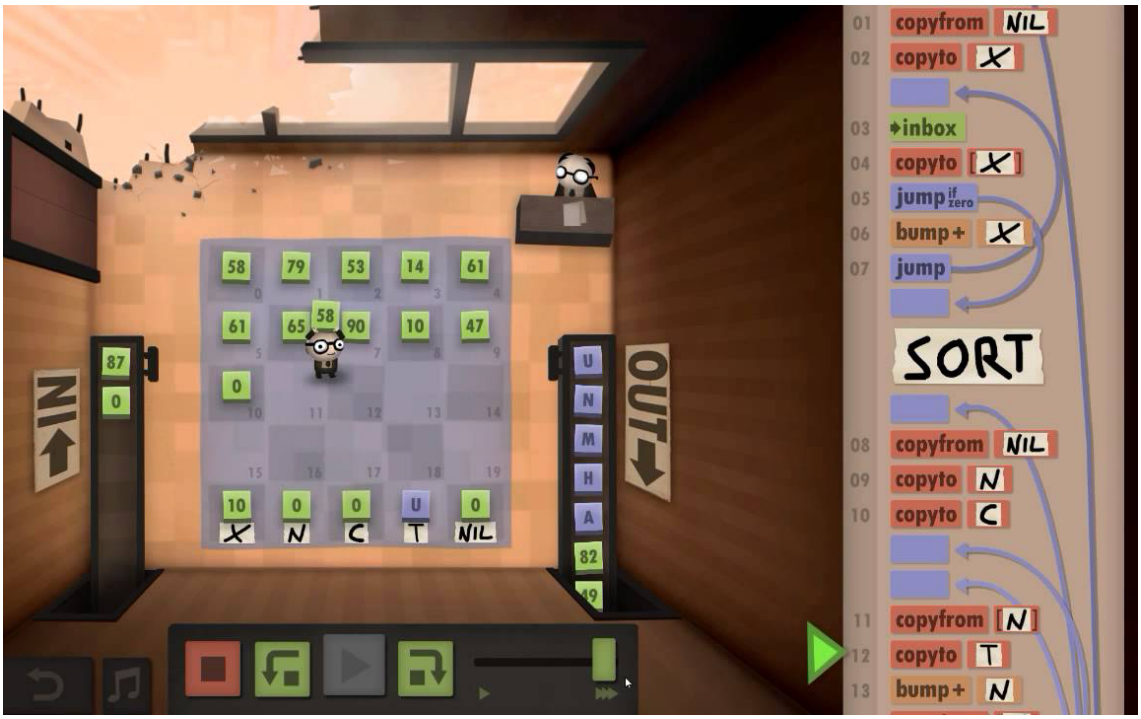

FIGURE 1 – Le jeu dont est tiré l'atelier (niveau 41)

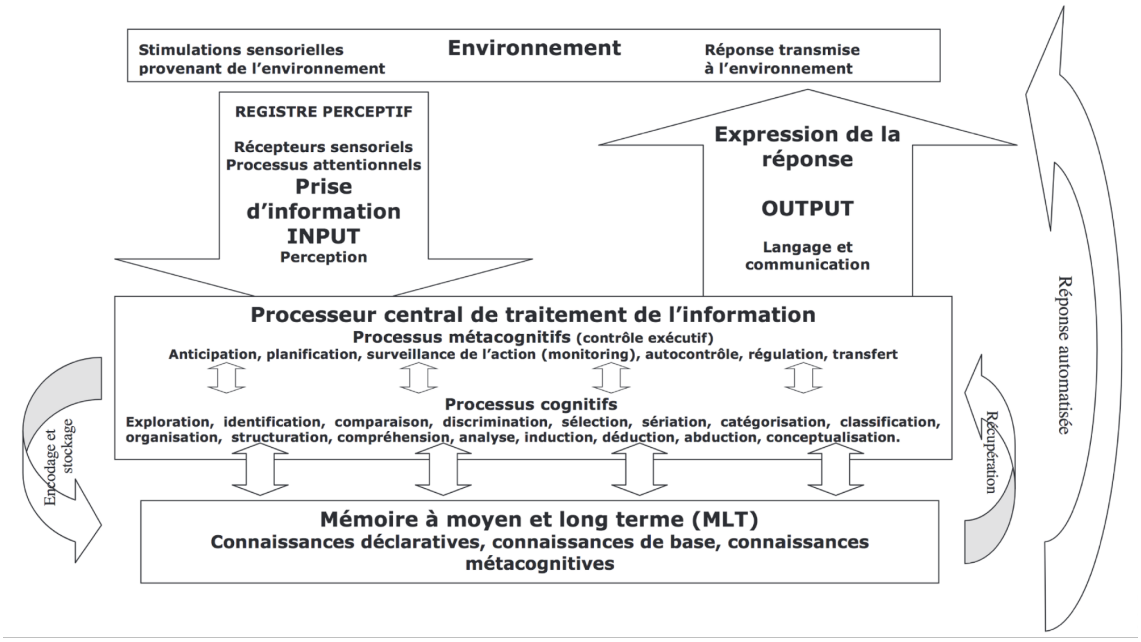

FIGURE 2 – Schéma du fonctionnement cognitif (Vianin, 2009)

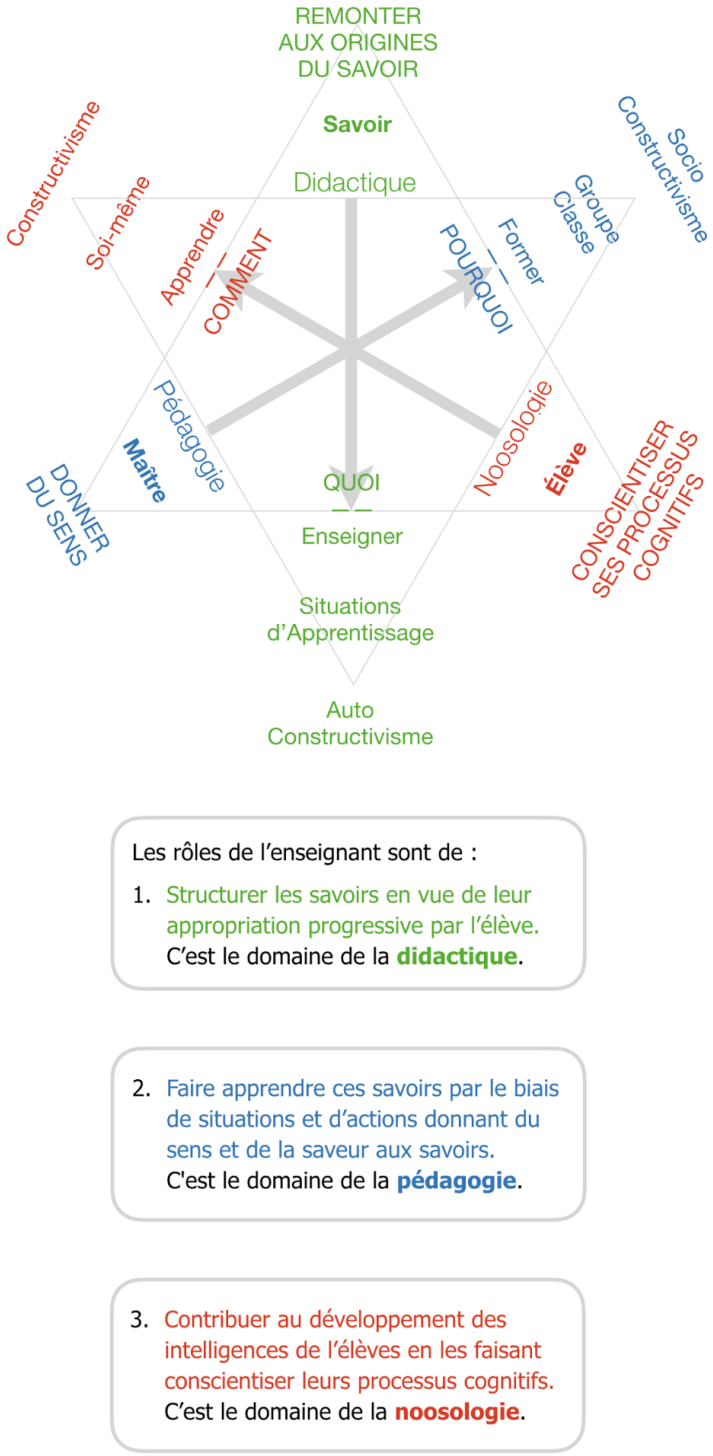

FIGURE 3 – Revisite du triangle de Houssaye