

## Construcción de modelos y simuladores con Squeak-Etoys como recurso de aprendizaje en la escuela media

Ricardo P. Salvador<sup>1</sup>, Claudia Pons<sup>2</sup>, Guillermo L. Rodríguez<sup>3</sup>

<sup>1</sup> Esc. Superior de Comercio "Libertador. Gral. San Martín" (U.N.R.), Balcarce 1240, 2000 Rosario, Santa Fe, República Argentina

<sup>2</sup> Laboratorio de Investigación y Formación en Informática Avanzada (U.N.L.P.), 50 y 120, La Plata, Buenos Aires, Argentina

<sup>3</sup> Fac. de Ciencias Exactas, Ingeniería y Agrimensura (U.N.R.), Av. Pellegrini 250, Rosario, Santa Fe, República Argentina

rsalvado@unr.edu.ar, cpons@info.unlp.edu.ar, guille@fceia.unr.edu.ar

**Resumen.** En este trabajo de investigación se estudia la influencia en el aprendizaje de cinemática de alumnos de secundaria utilizando la construcción de una simulación en el entorno de programación *Squeak-Etoys*. Al considerar *a)* los efectos de usar simulaciones como recurso educativo, *b)* el surgimiento de nuevos entornos de programación que combinan usabilidad, manipulación de multimedia y programación por bloques, y *c)* la revaloración en el ámbito educativo de las ciencias de la computación y la programación, hizo intuir que construir una simulación sería un recurso útil para el aprendizaje. Se realizó un trabajo experimental con dos grupos de alumnos de secundaria. Uno de los grupos utilizó un tutorial interactivo con indicaciones y recursos para programar la simulación. La experiencia se evaluó mediante un protocolo pre-test / post-test. Los resultados indican que los alumnos aumentaron las calificaciones y se interesaron en la programación como recurso didáctico. La recepción que la actividad tuvo en los alumnos anima a diseñar actividades similares y explorar su uso en otras disciplinas.

**Palabras clave:** aprendizaje, Física, educación media, simulación, programación.

### 1 Introducción

El objetivo de este trabajo es indagar la utilidad del diseño y desarrollo de modelos y simulaciones informáticas por parte de los alumnos como recurso de aprendizaje en la escuela media, en una disciplina distinta de las relacionadas a la informática misma, utilizando un entorno de programación amigable *Squeak-Etoys*. El aporte que se busca es proponer una metodología de diseño de actividades en la que los alumnos deban modelar y simular contenidos escolares.

En la sección 2, marco teórico, se explican las posibilidades que brinda la construcción de simulaciones como recurso didáctico, se menciona la Taxonomía de Bloom como marco de referencia para habilidades intelectuales, se recopilan aportes del construccionismo a la enseñanza mediante computadoras, se exponen referencias a la correlación entre programar y el desarrollo de pensamiento computacional, se destacan iniciativas de alcance nacional e internacional promoviendo las ciencias de la computación y la programación, y se caracteriza el entorno de programación *Squeak-Etoys*. El marco metodológico del trabajo de campo, los recursos y los instrumentos utilizados se exponen en la sección 3. En la sección 4, se describen los resultados: las características de la escuela donde se realizó el trabajo de campo, de los grupos de alumnos participantes, de los encuentros que conformaron la actividad experimental y los resultados de los diagnósticos y evaluaciones, previas y posteriores a la misma. En la sección 5 se exponen las conclusiones alcanzadas, y para finalizar, las perspectivas actuales.

## 2 Marco teórico

### 2.1 Computadoras, simulación y educación

Frente a tecnologías educativas que ponen al alumno en un rol relativamente pasivo, tales como el libro, el audio y el video, las computadoras como máquinas interactivas aportan la posibilidad de un aprendizaje especialmente activo, tema sobre el que contamos con notables aportes de Seymour Papert<sup>1</sup> y Alan Kay<sup>2</sup>.

Construir una simulación implica *a)* comprensión del sistema que se va a simular, *b)* abstracción de los elementos pertinentes a la modelización, *c)* puesta en juego de habilidades relacionados al pensamiento computacional, *d)* representar el conocimiento e interactuar con esa representación, involucrando el ensayo/error y la puesta a prueba de hipótesis, *e)* interactuar con otros (pares, profesores) en el desarrollo y en la comunicación de la actividad.

Esto es importante ya que el alumno (o grupo) realiza una tarea compleja e integral respecto del objeto de estudio, llevando a cabo un conjunto de habilidades cognitivas

---

<sup>1</sup> Fundador del Laboratorio de Inteligencia Artificial del Massachusetts Institute of Technology, y del Grupo de Epistemología y Aprendizaje del MIT-MediaLab, creador del lenguaje Logo, asesor en Squeak-Etoys, scratch en el proyecto One Laptop per Child.

<sup>2</sup> Investigador del Xerox Palo Alto Research Center, ARPANet, Atari, desarrollo de la primera computadora personal gráfica orientada a objetos, SmallTalk, Squeak-SmallTalk, Etoys, Hewlett Packard, presidente del Instituto Viewpoints Research.

que abarca desde las más simples a las de mayor complejidad intelectual, y desde las más pasivas a las que le exigen al alumno más compromiso intelectual.

## **2.2 Aportes del Construccinismo a la enseñanza mediante computadoras**

Seymour Papert creó el concepto de construccionismo, al que define como dar a los niños buenas cosas para hacer de manera que el aprendizaje tenga lugar por el hecho de hacerlas, ya que pensaba que en lugar de sólo interactuar con la tecnología deberían aprender a programar sus propias animaciones, juegos y simulaciones, y en el proceso, aprender habilidades para resolver problemas y estrategias para el diseño de proyectos [1].

Sostiene que el aprendizaje se da indirectamente, al utilizar el conocimiento y no sólo memorizándolo. En otras palabras, la forma en que mejor aprendemos es a través de la acción de construir algo externo a nosotros mismos: construir una torre, escribir un cuento, construir un artefacto robótico, programar un videojuego, o realizar una animación, son ejemplos de construcción. Este tipo de actividades tiene valor respecto de lo que Papert llama “prueba de la realidad”: si no funcionan, son un reto para comprender por qué, y para superar los obstáculos. Pueden compartirse y discutirse con otras personas. Y sirven como objetos de transición para la apropiación personal de estas ideas [2].

Al elaborar el enfoque construccionista, Papert sentó las bases de la enseñanza con tecnologías digitales reconociendo un mundo de cambios cada vez más rápidos, dando un lugar preferencial a los intereses del alumno, para el que los saberes nuevos deben tener sentido actual, valorando la confrontación de modos de pensar diversos y alternativos, dando a la programación un lugar clave, asumiendo que los alumnos pueden aprender a programar desde edades tempranas, donde se presentarán situaciones nuevas aún para el docente, que se transforma en un maestro que aprende a la par que sus alumnos y rescatando la capacidad de simulación de las computadoras como recurso educativo.

## **2.3 El pensamiento computacional**

El pensamiento computacional es la actividad intelectual relativa a la formulación de un problema de manera que pueda admitir una solución por medios computacionales, a la vez que la solución pueda ser llevada a cabo por un ser humano o una máquina, o de ambas maneras [3].

Kafai y Burke [4], señalan que la programación, hoy se reconoce como una respuesta a la necesidad de desarrollar pensamiento computacional en los estudiantes ya que éstos pueden resolver problemas, diseñar sistemas para la vida diaria, y generar progresos en otras disciplinas.

Resnick y Brennan [5], en estudios sobre el diseño de medios interactivos realizado por jóvenes, elaboraron un marco de referencia para el pensamiento computacional y cómo evaluar logros sobre el mismo; elaboraron una lista de siete conceptos computacionales comunes a muchos lenguajes de programación y presentes en el pensamiento computacional: *secuencia*, *ciclo*, *evento*, *paralelismo*, *condicional*, *operador*, y *dato*. También observaron prácticas de diseño relativas a cómo aprendían los participantes de sus estudios: *ser incremental e iterativo*, *ensayar y depuración*, *reusar y remezclar*, *abstraer* y *modularizar*. La conexión entre programación y pensamiento computacional parece corroborarse en otras fuentes consultadas [6] [7].

#### **2.4 Iniciativas promoviendo las Ciencias de la Computación y la Programación en las escuelas**

Actualmente se está revalorizando la enseñanza de las Ciencias de la Computación en el ámbito escolar y la enseñanza de la Programación y su uso como recurso educativo y como competencia general en distintos lugares del mundo. Por ejemplo, los proyectos *Raspberry Pi* (<http://www.raspberrypi.org/about/>), *Scratch* (<http://scratch.mit.edu/about/>), *Squeak-Etoys* (<http://squeakland.org/about/>), y Code.org (<http://code.org>).

En Argentina, la Ley de Educación Nacional N° 26206 (2006) promueve las competencias relativas las tecnologías de la información y la comunicación (capítulo 11, inciso *m*), y la enseñanza de la “Programación” es declarada de importancia estratégica por la 65° Asamblea del Consejo Federal de Educación (Resolución 263/15, art. 1°). *Program.AR* es una iniciativa de la Fundación Sadosky, surgida en 2012, y su objetivo es llevar el aprendizaje de Ciencias de la Computación a las escuelas (<http://program.ar/quienes-somos/>). *La hora del código* (iniciativa conjunta de *Program.AR* y *Code.org*) ofrece recursos para aprender a programar de forma amena y divertida como una forma de aproximarse a las Ciencias de la Computación.

#### **2.5 Squeak-Etoys: un entorno de programación diseñado para la educación**

*Squeak-Etoys* es un sistema de autoría multimedia especialmente dirigido a aprender ideas construyéndolas. Inspirado en LOGO, *Smalltalk*, *Hypercard* y *Starlogo*, se desarrolló a partir de *Squeak* en Walt Disney Imagineering (la rama de investigación y desarrollo de The Walt Disney Company) y luego fue apoyado por el Viewpoints Research Institute (VPRI) como uno de sus proyectos. *Etoys* presenta un estilo unificado, interfaz de usuario, medios de comunicación y entorno de programación para construir cosas con computadoras [8]. Integrado en el software de las laptops del proyecto One Laptop Per Child, esta aplicación multiplataforma (Linux, Windows y Mac) soporta objetos (texto, imágenes bitmap y vectores, sonidos) que pueden ser editados, sensar eventos, comunicarse con otros objetos, e incluir código de programa. Su simplicidad es proporcional a la posibilidad de obtener resultados de

gran calidad didáctica: facilitando la escritura de código, permite al usuario concentrarse más en el problema que debe resolver al implementar su simulación.

### 3 Marco metodológico

El trabajo de campo consistió en evaluar el efecto del uso del modelado y la simulación en la construcción de un programa como recurso para el aprendizaje, mediante un protocolo pre-test / post-test en el que participaron un grupo control y un grupo experimental. Se contactó a un docente de Física quien facilitó la participación dos de sus cursos, de 30 y 24 alumnos, en los roles de grupo control y grupo experimental. Los alumnos de este último grupo desarrollaron la simulación sobre un ejercicio de “encuentro” de Movimiento Rectilíneo Uniforme. En vínculo con esto, existen diversas experiencias de investigaciones utilizadas en la enseñanza de la Física [9].

Previamente a la realización del trabajo experimental, se realizó en ambos grupos *a)* un diagnóstico de los grupos control y experimental con el objeto de conocer características del grupo en cuanto al uso de recursos tecnológicos digitales, forma de abordar problemas, conocimiento sobre modelado y opinión de su utilidad en el aprendizaje, y *b)* una encuesta sobre contenidos de Cinemática, con preguntas sobre Movimiento Rectilíneo Uniforme.

La actividad experimental se realizó en el salón donde habitualmente tiene clase el grupo; se utilizaron netbooks del Plan Conectar Igualdad, que incluyen *Squeak-Etoys*. Se previó que el trabajo se hiciera en equipos, especialmente para favorecer la cooperación ante un recurso nuevo.

Se llevó a cabo utilizando un tutorial interactivo diseñado para ofrecer la guía para la actividad y el espacio de trabajo en una misma ventana, de manera que se evitara cambiar de ventanas continuamente. Las consignas y orientaciones para la actividad también se imprimieron para ofrecer una fuente alternativa, por si se experimentaba alguna dificultad en la lectura, o si se deseaba eliminar la guía del tutorial para contar con más espacio en la pantalla (sobre todo considerando que las pantallas de las netbooks son de sólo 10,1 pulgadas).

Debido a que los alumnos debían intervenir modificando y creando guiones y código, se esperaba que surgieran numerosos imprevistos, por lo que la guía fue redactada con la expectativa de que promoviera el trabajo autónomo y que la intervención del profesor estuviera dispuesta sobre todo para la observación del trabajo de los grupos e intervenir en situaciones emergentes.

El tutorial interactivo se desarrolló utilizando un recurso de *Squeak-Etoys* llamado “libro” que permite añadir páginas, en forma similar a una presentación multimedia, con la diferencia de que cada diapositiva también puede contener objetos con sus guiones y se puede intervenir la presentación mientras se está “leyendo”.

Se organizó en una serie de problemas parciales, en grado creciente de complejidad respecto del uso del entorno de programación a la vez que cada etapa progresa hacia la resolución del ejercicio construyendo etapas de la simulación.

## **4 Resultados**

### **4.1 La Escuela**

El trabajo de campo se realizó en la Escuela Superior de Comercio “Libertador General San Martín”, dependiente de la Universidad Nacional de Rosario. La Escuela ofrece la orientación “Comercialización”. Situada en el centro de la ciudad de Rosario, Santa Fe, tiene una población de aproximadamente 1200 alumnos agrupados en comisiones de treinta, que se distribuyen en dos turnos (mañana y tarde), cinco años y cuatro divisiones.

La estructura académica de la Escuela agrupa espacios curriculares afines en Departamentos encabezados por un Jefe y un Adscripto o Coordinador de Área, que coordina cada espacio curricular en particular.

El espacio curricular “Física” está presente de 3° a 5° año, con una carga horaria de 4 horas cátedra semanales; la unidad didáctica correspondiente a “Cinemática” es la primera de 4° año.

La Escuela cuenta con el espacio curricular “Informática” de 1° a 4° año, con una carga horaria de 2 horas cátedra semanales en cada año. Los contenidos no incluyen Programación ni Algoritmia.

### **4.2 Actividad experimental**

#### **4.2.1 Indicaciones previas**

Al principio de la primera de las tres clases que duró la actividad, se señalaron los siguientes criterios, aunque se recordaron en forma general y a cada grupo en particular, durante toda la experiencia, y cada vez que se consideró necesario: *a)* se destacó que habría cosas no explicadas, a propósito para que intuyan, imaginen y propongan (y lo pongan en práctica), o para que interactúen efectuando preguntas, *b)* se insistió en que la actividad funciona mejor con una actitud proactiva, por lo que se pidió a los alumnos que prueben, intenten, pongan a prueba este material y sus propias suposiciones, sin esperar hacer todo "correcto"; ni esperar a tener una idea brillante para poner manos a la obra: ¡no se va a romper nada! *c)* también se señaló

que la actividad contiene indicaciones en cada página del tutorial para que cada grupo pueda trabajar a su propio ritmo, y cuando lo necesite, por supuesto, llame al profesor.

#### **4.2.2 Desarrollo en el tiempo**

El inicio de la actividad experimental estaba previsto para fines de abril, se esperaba una duración de 2 a 3 módulos de 80 minutos, con una frecuencia semanal, para finalizarla. Por imprevistos que alteraron el calendario escolar, la actividad se desarrolló durante tres clases, el 9 de junio, y el 5 y el 12 de agosto de 2016:

##### **Primera clase**

Al comienzo había aún 3 netbooks para desbloquear (aunque todos los días de la semana previa se acudió al curso para desbloquear las netbooks, y solicitárselo a los que aún las tuvieran bloqueadas). Para el desarrollo de la actividad se requería cargar el tutorial interactivo (un archivo de 1,1 Mb), por lo que se previó descargarlo de Internet, o desde dos pendrives, opción que se utilizó ya que no se disponía de WiFi y la señal de telefonía móvil no era suficiente para implementar un punto de acceso desde un celular. Se contaba con 10 netbooks, y la tarea de copiar los tutoriales insumió casi 40 minutos de los 80 de la clase, en parte, debido a que algunas máquinas no reconocían los pendrives (o tardaban mucho tiempo en hacerlo) y otras directamente no tenían la configuración de plataformas original y hubo que instalar *Etoys*.

En los poco más de 40 minutos en que se trabajó en la actividad, los 10 grupos que se formaron trabajaron evidenciando interés, con algunas distracciones pero superando más del 50% de la actividad.

Este encuentro incluyó la presentación de la actividad, sus objetivos, la forma de trabajar y el entorno de programación que se usó, sobre el cual el tutorial incluye indicaciones.

##### **Segunda clase**

Se continuó el trabajo utilizando una versión modificada del proyecto, en la que se realizaron correcciones en la redacción.

En esta ocasión, la mayoría de los grupos llegó a la instancia del trabajo en que se simula completamente el comportamiento de los autos “protagonistas” de la simulación. Como los dos autos que intervienen en el ejercicio responden al mismo modelo de comportamiento, una vez terminado uno de los autos, sólo es necesario duplicarlo y modificar su nombre, los parámetros relativos a velocidad, a tiempo, y el aspecto (esto último, para diferenciarlo visualmente del primer auto). Casi todos, a la hora de modificar el aspecto (uno de los autos del ejercicio es azul y el otro es verde) hicieron un paréntesis para cambiar algo más que el color a su nuevo auto.

##### **Tercera clase**

La mayoría de los grupos resolvió los desafíos de la penúltima diapositiva, en que simula la situación planteada en el ejercicio; sin embargo no todos terminaron la actividad de la última diapositiva, donde debían obtener y mostrar las respuestas al ejercicio sobre hora y distancia del encuentro entre los dos autos. Al llegar a la mitad

de la clase, se realizó una evaluación del trabajo de cada grupo y al observar que la mayoría encontraba dificultades en los dos últimos desafíos se resolvió explicarlos a todo el grupo utilizando un proyector, para que posteriormente cada grupo intentara terminar su trabajo.

En las clases experimentales se evidenció motivación y se manifestaron inquietudes acerca de cómo cambiar otros aspectos de la simulación, que en principio no estaba previsto como parte de la actividad: *a*) cómo modificar el aspecto (colores, formas) de la simulación, *b*) simular otro tipo de movimiento (realizar cambios de dirección, describir curvas, por ejemplo), *c*) también surgieron consultas sobre la posibilidad de crear simulaciones en otros espacios curriculares, sobre el entorno de programación utilizado y sobre cómo se aprende a programar. Se observó que el aprendizaje de la interface de usuario, al inicio de la experiencia, insumió una cantidad de tiempo que disminuyó en las siguientes clases.

### **4.3 Resultados de las encuestas**

#### **4.3.1 Grupo control**

En este grupo, compuesto por un 62% de alumnas y 38% de alumnos, el 70,37% manifestó en el pre-test que había usado software en la escuela para representar situaciones que habitualmente se hacen con lápiz y papel, el 48,15% para resolver problemas y el 18% para construir aplicaciones. El 88,88% afirmó usar tecnología informática para resolver problemas cotidianamente y el 55,56% prefiere resolver problemas en forma grupal.

Frente a la pregunta sobre el concepto de simulación, en el pre-test el 29,63% contestó correctamente, mientras que en el post-test lo hizo el 46,67%. Respecto del concepto de modelo, en el pre-test contestó correctamente el 18,52% y el 40% lo hizo en el post-test. Se observó en el post-test una reducción de la incertidumbre respecto a estos conceptos, de 37,04% a 16,67% y de 40,74% a 6,67% respectivamente. El 77,78%, en el pre-test, contestó afirmativamente acerca de si el uso y/o construcción de modelos y/o simulaciones podría contribuir a sus aprendizajes escolares, y en el post-test lo hizo el 76,67%. En la evaluación de contenidos de Cinemática, en el pre-test, el promedio de respuestas correctas fue de 59,26%, mientras que en el post test, fue de 57,06%.

#### **4.3.2 Grupo experimental**

El grupo experimental estuvo compuesto por un 79% de alumnas y 21% de alumnos, de los cuales en el pre-test, el 54,17% manifestó haber usado software en la escuela para representar situaciones que habitualmente se hacen con lápiz y papel, y el 37,5% para resolver problemas. El 66,67% afirmó usar tecnología informática para resolver problemas cotidianamente y el 70,83% prefiere resolver problemas en forma individual.



En cuanto a la pregunta sobre el concepto de modelo, el 25% contestó correctamente en el pre-test, mientras que en el post-test lo hizo el 36,36%. Respecto del concepto de simulación, las respuestas correctas en el pre-test fueron el 33,33%, siendo el 59,59% en el post-test. Se observó en el post-test una reducción de la incertidumbre respecto a estos conceptos, de 70,83% a 33,33% y de 58,33% a 8,33% respectivamente. En la evaluación pre-test, acerca de si el uso y/o construcción de modelos y/o simulaciones podría contribuir a sus aprendizajes escolares, el 54,17 contestó afirmativamente y alcanzando al 72,73% estas respuestas en el post-test. En la evaluación de contenidos de Cinemática, en el pre-test, el promedio de respuestas correctas fue de 56,25%, mientras que en el post test, fue de 60,42%.

## 5 Conclusiones

La construcción de una simulación mediante *Squeak-Etoys* favoreció el aprendizaje de contenidos de Cinemática, evidenciado en el aumento en las calificaciones de las evaluaciones post-test. Los resultados de la actividad experimental fueron satisfactorios aún tratándose de un grupo heterogéneo.

Programar una simulación se mostró como un recurso didáctico válido aún para alumnos sin conocimiento de programación ni algoritmia. Tras realizar la actividad experimental, los alumnos que comprendían el concepto de simulación aumentó casi tres veces.

Construir la simulación del ejercicio de Cinemática generó motivación en los alumnos, así como interés y expectativas positivas respecto de usar la misma forma de trabajo en otros aprendizajes.

Los alumnos, además de valorar el tipo de actividad que realizaron, demostraron curiosidad por conocer modos de expresarse y poner su sello personal en la actividad usando las herramientas del entorno de programación utilizado.

La actividad presentó numerosos imprevistos y desafíos, lo que significó mayor intensidad en la tarea a la vez que promovió una flexibilidad en la asimetría entre los roles docente y alumno, al encarar juntos esos desafíos.

El trabajo en torno al tutorial interactivo deja una serie de recursos reutilizables en nuevas actividades, y retroalimentación para mejorarlas, reduciendo el tiempo de nuevos desarrollos.

Proponer la experiencia con características lúdicas, desafíos, contenidos significativos para los alumnos y mediatizada por las nuevas tecnología, favoreció el aprendizaje y el interés de los alumnos en los medios utilizados, siendo nuestro desafío actual formalizar su utilización dentro de la escuela.

## 6 Bibliografía

1. Resnick, M. (2012), "Reviving Papert's Dream", "Educational-technology – the magazine for managers of change in education", Volume 52, Number 4, July/August 2012, <http://web.media.mit.edu/~mres/papers/educationaltechnology2012.pdf>
2. Papert, S. (1999), "Logo Philosophy and Implementation", Logo Computer Systems Inc. Introduction.
3. Wing, J. M. (2010), "Computational Thinking: What and Why?", Scholl of Computer Science, Carnegie Mellon University, <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
4. Kafai, Y. and Burke, Q. (2014), "Connected Code – Why Children Need to Learn Programming", The MIT Press
5. Resnick, M. and Brennan, K., (2012), "New frameworks for studying and assessing the development of computational thinking", [http://web.media.mit.edu/~kbrennan/files/Brennan\\_Resnick\\_AERA2012\\_CT.pdf](http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf)
6. Lye, S.Y. and Koh, J.H.L. (2014), "Review on teaching and learning of computational thinking through programming: What is next for K-12? ", artículo en "Computers in Human Behavior", Dec2014, Vol. 41, p51-61. 11p. URL: <http://www.sciencedirect.com/science/article/pii/S0747563214004634>
7. Yohan Hwang, Kongju Mun, Yunebae Park (2016), "Study of Perception on Programming and Computational Thinking and Attitude toward Science Learning of High School Students through Software Inquiry Activity: Focus on using Scratch and physical computing materials", Journal of the Korean Association for Science Education, URL: [http://koreascience.or.kr/article/ArticleFullRecord.jsp?cn=GHGOBX\\_2016\\_v36n2\\_325](http://koreascience.or.kr/article/ArticleFullRecord.jsp?cn=GHGOBX_2016_v36n2_325)
8. Kay, Alan (2007) "Children Learning By Doing – Etoys on the OLPC XO", Viewpoints Research Institute, <http://wiki.laptop.org/images/2/28/OLPCEtoys.pdf>
9. Taub, Armoni, Bagno, Ben-Ari (2015), "The effect of computer science on physics learning in a computational science environment", Computers & Education Volume 87, September 2015, Pages 10–23, URL: <http://www.sciencedirect.com/science/article/pii/S0360131515000913>