

Simulaciones con Squeak-Etoys para plantear y resolver problemas

Experiencia de programación como medio para plantear y resolver un problema de Cinemática en escuela media

Resumen - En este trabajo se estudia la influencia, en el aprendizaje de cinemática de alumnos de secundaria, de la construcción de una simulación mediante Squeak-Etoys. El origen fue la consideración de a) el potencial de las simulaciones como recurso educativo, b) nuevos entornos de programación que reúnen gran usabilidad, gestión de multimedia y pseudocódigo traducido a numerosas lenguas c) la revaloración en el ámbito educativo de las ciencias de la computación y la programación, factores que hicieron pensar que construir una simulación sería un recurso útil para el aprendizaje. Se realizó un trabajo experimental utilizando un protocolo pre-test / post-test, con dos grupos de alumnos. Los resultados indican que los alumnos aumentaron las calificaciones, se interesaron en la programación como recurso didáctico, y la recepción que la actividad tuvo en los alumnos anima a diseñar actividades similares y explorar otras disciplinas.

Simulación y programación para representar y resolver problemas

Las computadoras, como herramientas de producción multimedial interactiva, aportan la posibilidad de un aprendizaje especialmente activo. Seymour Papert, en su enfoque constructorista, sentó las bases de la enseñanza con tecnologías digitales dando a la programación un lugar clave, y rescatando la capacidad de simulación de las computadoras como recurso educativo. Consideraba que la mejor forma de aprender es construyendo algo externo a nosotros mismos: construir una maqueta, escribir un cuento, construir un artefacto robótico, programar un videojuego. 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 (Papert, 1999).

La construcción de una simulación implica: a) comprender el sistema que se va a simular, b) abstraer los elementos pertinentes a la modelización, c) poner en juego habilidades relacionados al pensamiento computacional y d) representar el conocimiento e interactuar con esa representación.

La revalorización actual de la enseñanza de las Ciencias de la Computación y la Programación en el ámbito escolar como recurso educativo y como competencia general en distintos lugares del mundo se ha concretado, por ejemplo, en los proyectos Raspberry Pi (www.raspberrypi.org/about/),

Scratch (scratch.mit.edu/about/), Squeak-Etoys (squeakland.org/about/) y Code.org. En Uruguay el Plan Ceibal es toda una referencia y pionero en América Latina.

Squeak-Etoys: programación a la medida de alumnos y maestros

Squeak-Etoys, basado en ideas de Alan Kay y Seymour Papert, y desarrollado en el área de investigación y desarrollo de The Walt Disney Company, es un sistema de autoría multimedia especialmente dirigido a alumnos, maestros y legos en general. 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.

Poniendo a prueba estas ideas

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 que dos autos pasan por una misma ciudad, en la misma dirección, con una diferencia de algunas horas: la consigna es averiguar cuándo el segundo auto, que se mueve más rápido, encuentra al primero, y a cuántos kilómetros de la ciudad de referencia. Si bien los alumnos cuentan con educación en Informática, ésta no incluye algoritmia ni programación.

Previa y posteriormente a la realización del trabajo experimental, ambos grupos realizaron a) un diagnóstico con el objeto de conocer 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 utilizando *netbooks* con Squeak-Etoys. Los alumnos se agruparon en equipos, especialmente para favorecer la cooperación ante un recurso nuevo. Se utilizó un tutorial interactivo diseñado para ofrecer la guía para la actividad y el espacio de trabajo en una misma ventana, enfocarse en ésta y evitar el cambio de ventanas continuamente. Las consignas y orientaciones para la actividad también se imprimieron para ofrecer una fuente alternativa. 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 (es decir, escribir código) a la vez que se está "leyendo". Algo así como arreglar el motor de un auto... ¡sin detenerlo!

Este tutorial 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.

Actividad experimental

Indicaciones previas

Antes de iniciar la actividad y durante el desarrollo de la misma se destacó que: *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 funcionaría 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 a no esperar hacer todo "correcto"; ni esperar a tener una idea brillante para ponerse 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.

Desarrollo en el tiempo

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 entre junio y agosto de 2016.

Primera clase. 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. 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 simulan 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.

Conclusiones

- La construcción de una simulación mediante Squeak-Etoys favoreció el aprendizaje de contenidos de Cinemática.
- Los resultados son favorables en coincidencia con otras investigaciones similares con eje en contenidos de Física (Taub y otros, 2015), y con experiencias similares en que también se ha recurrido a entornos de trabajo amistosos para el usuario (Yohan, 2016).
- Programar una simulación se mostró como un recurso didáctico válido aún para alumnos sin conocimiento de programación ni algoritmia, y tratándose de un grupo heterogéneo.
- Tras la actividad experimental, los alumnos afianzaron el concepto de simulación.
- La actividad presentó numerosos imprevistos y desafíos para el docente, 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, generó motivación en los alumnos, así como interés y expectativas positivas respecto de usar la misma forma de trabajo en otros aprendizajes.
- En los primeros pasos en programación se dan mejores resultados usando interfaces gráficas amigables similares a Scratch y Squeak-Etoys, que fue el entorno utilizado en el trabajo de campo.
- 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 modificando distintos aspectos del tutorial, más allá de las consignas.

Autores

- Ricardo Pablo Salvador, Esc. Superior de Comercio "Libertador. Gral. San Martín" (U.N.R.), Balcarce 1240, 2000 Rosario, Santa Fe, República Argentina, rsalvado@unr.edu.ar
- Claudia Pons, Laboratorio de Investigación y Formación en Informática Avanzada (U.N.L.P.), 50 y 120, La Plata, Buenos Aires, Argentina, cpons@info.unlp.edu.ar
- Guillermo L. Rodríguez, Fac. de Ciencias Exactas, Ingeniería y Agrimensura (U.N.R.), Av. Pellegrini 250, Rosario, Santa Fe, República Argentina, guille@fceia.unr.edu.ar

Bibliografía

1. Kafai, Y., Burke, Q. (2014), "Connected Code – Why Children Need to Learn Programming", The MIT Press
2. Kay, Alan (2007) "Children Learning By Doing – Etoys on the OLPC XO", Viewpoints Research Institute, <http://wiki.laptop.org/images/2/28/OLPCEtoys.pdf>
3. Lye, S.Y., 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>
4. Papert, Seymour (1999), "Logo Philosophy and Implementation", Logo Computer Systems Inc. Introduction.
5. Resnick, Mitchel (2012), "Reviving Papert's Dream", "Educational-technology – the magazine for managers of change in education", Volume 52, Number 4, JulyAugust 2012, <http://web.media.mit.edu/~mres/papers/educationaltechnology2012.pdf>
6. Resnick, M., 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
7. 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>

9. 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=GHHOBX_2016_v36n2_325