

Eficacia Pedagógica de los Niveles de Implicación de los Estudiantes con las Animaciones de Programas y Algoritmos

Jaime Urquiza Fuentes

Departamento de Lenguajes y Sistemas Informáticos I,
Universidad Rey Juan Carlos,
C/ Tulipán s/n, 28933 Móstoles, Madrid
jaime.urquiza@urjc.es

Resumen. Las animaciones de programas y algoritmos son representaciones visuales, visualizaciones, dinámicas de su comportamiento durante la ejecución. Las animaciones se han utilizado en ámbitos educativos, pero no existen resultados empíricos que soporten en su totalidad la creencia de que ayudan a aprender mejor. El resultado más significativo sobre este tema concluye que: es más importante lo que los estudiantes hacen con las animaciones, que lo que estas les muestran. En este capítulo se presenta un estudio sobre la eficacia pedagógica de los diferentes usos de las animaciones en términos de la implicación de los estudiantes con estas. Aunque es necesario un análisis más profundo contando con experiencias fallidas, hemos detectado la existencia de ciertas características que podrían ser interesantes a la hora de diseñar experiencias educativas con animaciones de programas.

Palabras clave: Visualización de programas, Visualización de algoritmos, Informática educativa, Implicación de los estudiantes.

1 Introducción

Los estudios realizados sobre la eficacia pedagógica de las animaciones de programas y algoritmos¹ no han dado resultados claros. Hasta la fecha, el resultado más significativo es el de Hundhausen et al. [11], donde se concluye que el esfuerzo de los estudiantes dedicado a las tareas relacionadas con las animaciones es más importante que los contenidos visuales mostrados por estas. También identificaron áreas sin suficientes resultados empíricos, como el uso de contenidos narrativos y explicaciones textuales integradas en las animaciones.

Siguiendo esta idea de ir mas allá de la visión pasiva de las animaciones, Naps et al. [20] desarrollaron una taxonomía donde se identificaban diferentes formas de

¹ De ahora en adelante utilizaremos indistintamente los términos “visualizaciones”, “animaciones” o “animaciones de programas y algoritmos” para designar la misma cosa: representaciones visuales, normalmente dinámicas, de programas o algoritmos.

2 Actas de SITIAE 2008

interacción de los estudiantes con las animaciones. La llamaron *Taxonomía de Niveles de Implicación*, y sugirieron una estructura jerárquica, donde la eficacia pedagógica de las animaciones mejoraría cuanto mayor fuera la implicación de los estudiantes con las animaciones.

Después de revisar la literatura sobre animaciones de programas y algoritmos, e investigar sobre algunos niveles de implicación [22,23], creemos que las mejoras educativas podrían depender de otras características además de la implicación. Esta revisión estudia los posibles efectos de estas características.

El resto del capítulo se organiza como sigue. En la sección 2 describimos el estudio realizado, el tipo de experimentos que hemos tenido en cuenta y las características que hemos revisado. A continuación, en la sección 3, detallamos experimentos estudiados según los niveles de implicación relacionados. En la sección 4 analizamos estos experimentos desde tres puntos de vista distintos. Y finalmente, en la sección 5 exponemos nuestras conclusiones, así como las líneas de trabajo futuro.

2 Descripción del estudio

La literatura sobre animaciones de programas y algoritmos con fines educativos es muy amplia, nosotros nos hemos ceñido a los experimentos sobre el uso de animaciones que han detectado alguna mejora educativa —adquisición de conocimiento, actitud hacia la asignatura o los materiales usados, o habilidad en tareas de programación—.

Si echamos un vistazo a los experimentos publicados, se puede suponer que las visualizaciones por sí solas no son suficientes para obtener mejoras educativas. De hecho, uno de los estudios más significativos del área [11] concluye que la forma en que los estudiantes usan las animaciones es más importante que lo que las animaciones muestran a los estudiantes. También existen experimentos exitosos donde se usan materiales de alta calidad junto con las visualizaciones [7], interfaces avanzadas para la manipulación de las visualizaciones [5], o se usan clases extra de trabajo con visualizaciones [19].

Nuestro objetivo es profundizar en el efecto que tienen estas características adicionales en las mejoras educativas detectadas. Dichas características son:

- **Contenidos narrativos y explicaciones textuales.** Estos pueden ayudar a los estudiantes a entender las representaciones gráficas generadas por los sistemas de visualización. Además, cuando los estudiantes construyen sus

propias animaciones junto con explicaciones textuales realizan un ejercicio de reflexión que podría mejorar el aprendizaje.

- **Retroalimentación sobre acciones de los estudiantes.** Durante la reproducción de una animación se puede hacer preguntas a los estudiantes de forma que deban predecir estados de ejecución futuros de los programas o algoritmos. La retroalimentación a sus respuestas podría reforzar las correctas y corregir las erróneas. Debido a que las animaciones proporcionan retroalimentación de forma inherente en el siguiente paso de la animación, nosotros sólo tendremos en cuenta la retroalimentación explícita, tanto para respuestas correctas como incorrectas.
- **Tiempo de trabajo adicional con animaciones.** Muchas de las tareas a realizar en un entorno educativo no se pueden sustituir por otras relacionadas con las animaciones, de forma que para usar las animaciones se necesita un tiempo extra.
- **Características avanzadas.** Algunos sistemas proporcionan contenidos avanzados, como mostrar diferentes comportamientos de un algoritmo, interfaces avanzadas para manipular las visualizaciones, o una integración avanzada con el entorno de programación.

Obviamente, también hemos tenido en cuenta las mejoras educativas detectadas por cada experimento, así como el nivel de implicación correspondiente.. Las mejoras educativas se pueden detectar como conocimientos adquiridos por los estudiantes, su habilidad para programar sus propias soluciones, o su actitud hacia las asignaturas o los materiales didácticos utilizados (normalmente, la adquisición de conocimiento está influenciada por la actitud de los estudiantes). El diseño de los experimentos contempla tanto el estudio de mejoras en uno o varios niveles de implicación en conjunto, como estudios comparativos entre diferentes niveles.

3 Experimentos estudiados

Hemos considerado 24 experimentos en total. En esta sección los describimos agrupados según el nivel de implicación donde se han detectado las mejoras educativas. La taxonomía define seis niveles de implicación, el primero denota la ausencia de uso de animaciones, *Sin Visión*, los cinco siguientes son: *Visión*, *Respuesta*, *Cambio*, *Construcción* y *Presentación*. La tabla 1 resume estos experimentos.

3.1 Nivel de implicación *Visión*

“Visión” puede considerarse como la forma básica de implicación, (...) un estudiante puede ver una animación de forma pasiva, pero también puede controlar el sentido de su reproducción o su temporización, usar varias ventanas (cada una con una vista diferente), o usar contenidos textuales o explicaciones en audio. (...) Las siguientes cuatro categorías incluyen al nivel de Visión. (Traducido de Naps et al. [20])

Los seis experimentos relacionados con este nivel han detectado mejoras educativas en términos de adquisición de conocimientos. El experimento descrito en el séptimo capítulo de la tesis doctoral de A. Lawrence [17] detectó mejoras cuando se usaban etiquetas textuales en las animaciones. El resto de experimentos de este nivel hacen estudios comparativos con el nivel *Sin visión*.

Crosby y Stelovsky [4] detectaron mejoras utilizando materiales didácticos multimedia compuestos por visualizaciones y contenidos narrativos.

Kann et al. [13] compararon los niveles de implicación *Sin visión*, *Visión*, *Construcción* y la unión de *Visión* y *Construcción*. Sin embargo, sólo detectaron mejoras entre los dos primeros. Además este es el único estudio de *Visión* donde no se usan narraciones o contenidos textuales.

Kehoe et al. [14] estudiaron el uso de animaciones simulando su utilización a modo de deberes para casa. Así, los estudiantes utilizaron las animaciones para responder a ejercicios sin ningún límite de tiempo.

El experimento de Kumar [15] representa un uso auxiliar de las visualizaciones. El objetivo principal de este sistema es la tutorización de los estudiantes proporcionándoles problemas generados automáticamente. En este experimento se detectó que el uso de visualizaciones en la retroalimentación proporcionada por el tutor mejoraba la adquisición de conocimientos de los estudiantes.

Finalmente, Urquiza [22] investigó los efectos de sustituir parte de sesiones de ejercicios de programación por sesiones donde los estudiantes consultaban animaciones con contenidos textuales integrados. Esta sustitución produjo mejoras educativas a largo plazo (las sesiones se desarrollaron durante medio curso de programación y la evaluación se hizo según las calificaciones en el examen de la asignatura).

3.2 Nivel de implicación *Respuesta*

“Respuesta”. La actividad clave de esta categoría es responder a preguntas sobre la visualización presentada por el sistema. (...) En el nivel de implicación de *Respuesta*, el estudiante usa la visualización como un recurso para responder preguntas. (Traducido de Naps et al. [20])

Los tres experimentos contemplados comparan este nivel con el nivel *Sin visión*. Los dos primeros experimentos detectaron mejoras educativas en términos de adquisición de conocimientos y las animaciones integraban contenidos narrativos adicionales. Aunque Byrne et al. [3] usaron animaciones que no integraban preguntas, el profesor las proporcionaba según los estudiantes iban viendo la animación. El sistema usado por Grissom et al. [6] sí que integraba automáticamente las preguntas dentro de la propia animación.

Finalmente, Laakso et al. [16] fueron más allá de simples preguntas, pidiendo a los estudiantes que simularan el comportamiento de distintos algoritmos. En este caso, los estudiantes manipulaban directamente las estructuras de datos simulando las modificaciones que el algoritmo debería realizar sobre ellas, y recibiendo retroalimentación explícita de sus simulaciones. Además, utilizaron el nivel de *Visión*, puesto que permitían a los estudiantes consultar animaciones relacionadas con el algoritmo a simular.

3.3 Nivel de implicación *Cambio*

“Cambio”, implica modificar la visualización. El ejemplo típico de estas modificaciones es permitir al alumno cambiar los datos de entrada del algoritmo que estudia, de forma que puede explorar el comportamiento del algoritmo en diferentes casos. (traducido de Naps et al. [20])

Los dos primeros experimentos utilizaron de forma conjunta los niveles *Respuesta* y *Cambio*, comparándolos con los niveles *Visión* y *Sin visión*, estudios I, II, IV y V de Hansen et al. [7]. En vez de utilizar animaciones aisladas con añadidos, Hansen et al. [7] produjeron materiales didácticos de alta calidad, proporcionando a los estudiantes tres tipos diferentes de animaciones de cada algoritmo — conceptual/abstracta, detallada y con gran volumen de datos de entrada—, a las que añadían preguntas con retroalimentación explícita.

6 Actas de SITIAE 2008

Lawrence estudió el efecto de permitir el cambio de los datos de entrada de los algoritmos en comparación con los niveles *Sin visión* y *Visión*. En el primer estudio —comparación con el nivel *Sin visión*— Lawrence et al. [18] detectaron mejoras educativas en términos de adquisición de conocimiento; las animaciones utilizadas tenían contenidos narrativos y los estudiantes que trabajaron con estas animaciones dispusieron de una sesión extra de laboratorio. En el segundo estudio —comparación con el nivel *Visión*— Lawrence [17] también detectó mejoras en la adquisición de conocimientos, pero esta vez sin características adicionales en el caso de las animaciones.

Ben-Bassat et al. [2] estudiaron el uso de una herramienta de visualización para enseñar Java a estudiantes noveles. Sus resultados muestran que sólo los estudiantes medios mejoraron su conocimiento del lenguaje, los buenos estudiantes no necesitaban la herramienta mientras que a los de bajo nivel la herramienta les parecía demasiado complicada. Moskal et al. [19] se centraron en estudiantes noveles con alto riesgo de suspender su primer curso de programación. Para ello impartían una asignatura extra donde se trabajaba con una herramienta avanzada de programación orientada a objetos. Los estudiantes de alto riesgo que cursaron esta asignatura mejoraron en cuanto a adquisición de conocimiento a los que no la cursaron, estos últimos también de alto riesgo.

Ahoniemi y Lahtinen [1] compararon este nivel con el nivel *Sin visión*. Los estudiantes debían resolver problemas en su propio tiempo de estudio, por lo que no existió límite de tiempo alguno, utilizando animaciones que integraban contenidos narrativos. Detectaron mejoras educativas en adquisición de conocimientos en los estudiantes que usaron las animaciones.

El último experimento del nivel de *Cambio* [5,12] detectó mejoras en la habilidad para programar con respecto al nivel *Sin visión*. Ambos grupos de estudiantes trabajaron con el mismo entorno de programación. Los que trabajaron con animaciones disponían de una versión especial que incluía visualizaciones avanzadas integradas en el propio entorno. Las tareas que realizaron los estudiantes consistieron en programación y depuración usando el propio entorno.

3.4 Nivel de implicación *Construcción*

“Construcción”. En este nivel de implicación, los estudiantes construyen sus propias visualizaciones sobre los algoritmos que están estudiando. Hundhausen y Douglas [27] identificaron dos formas de construir las visualizaciones: generación directa y construcción manual. (...) Es importante recordar que este

nivel de implicación no conlleva necesariamente la codificación del algoritmo que se visualiza. (traducido de Naps et al. [20])

Stasko [21] diseñó ejercicios de clase donde los estudiantes tenían que construir sus propias animaciones, lo que incluía actividades propias del nivel de *Cambio*. Detectó que los estudiantes dedicaron más tiempo a estudiar aquellos algoritmos para los que construyeron animaciones.

Urquiza y Velázquez [23] hicieron un estudio comparativo a corto plazo con el nivel de *Visión*. Los estudiantes del grupo de *Construcción* generaron animaciones con explicaciones textuales integradas usando un enfoque que requiere poco esfuerzo de construcción mientras que el resto sólo vieron animaciones del mismo tipo, pero construidas por los profesores de la asignatura. Detectaron mejoras en adquisición de conocimientos y actitud de los estudiantes, los “constructores” estudiaron el algoritmo durante más tiempo que los “visores”.

Finalmente, Urquiza [22] estudió el efecto de las mismas tareas de construcción de animaciones en una evaluación a largo plazo, comparando tres niveles distintos *Construcción*, *Visión* y *Sin visión*. Detectó mejoras en la actitud de los estudiantes constructores con respecto a los otros dos niveles. También detectó mejoras en adquisición de conocimientos en la comparación de *Construcción* y *Sin visión*.

3.5 Nivel de implicación *Presentación*

“Presentación”, supone presentar una visualización a una audiencia, pudiendo recibir retroalimentación así como celebrar debates. (traducido de Naps et al. [20])

Los tres experimentos que estudian este nivel también incluyen tareas de construcción de animaciones, por lo tanto integran contenidos narrativos. Dos de ellos se centran en la conjunción de estos niveles [9,10], mientras que el otro experimentó comparó esta conjunción de niveles con el nivel de *Visión*.

En primer lugar, Hundhausen [9] estudió el uso de diferentes herramientas para construir animaciones y su posterior presentación. En concreto usó una herramienta muy conocida de visualización de algoritmos comparándola con otras herramientas seleccionadas por los propios estudiantes —transparencias electrónicas, acetatos e incluso dibujos a mano—. Este estudio detectó mejoras en la actitud de los estudiantes que eligieron sus propias herramientas. Usando estos resultados diseñó

una herramienta de construcción de algoritmos y la comparó de nuevo con herramientas seleccionadas por los propios estudiantes [10]. En este caso, detectaron mejoras en la habilidad en tareas de programación en aquellos estudiantes que habían utilizado la herramienta diseñada.

Finalmente, Hübscher-Younger y Narayanan [8] compararon los niveles de *Presentación* y *Construcción* con *Visión*. Animaron a los estudiantes a generar animaciones —esta tarea era voluntaria— y les pidieron que evaluaran aquellas generadas por sus compañeros —esta tarea era obligatoria—. Los estudiantes pudieron elegir las herramientas de construcción de animaciones. Detectaron mejoras educativas en términos de adquisición de conocimientos de los estudiantes que además de evaluar las animaciones generaron las suyas propias.

4 Discusión

En esta sección analizamos los resultados obtenidos por los experimentos descritos anteriormente. Primero lo haremos desde una visión global. Después damos un conjunto de recomendaciones, agrupadas por nivel de implicación, para diseñar experiencias educativas provechosas con animaciones. Finalmente, sugerimos posibles mejoras que se podrían obtener cambiando el nivel de implicación utilizado.

4.1 Una visión global de los experimentos

Claramente se ve que las animaciones pueden ayudar a mejorar el aprendizaje. El 75% (18/24) de los experimentos han detectado mejoras en términos de adquisición de conocimientos, junto con más del 20% (5/24) que detectaron mejoras de actitud hacia las asignaturas o los materiales didácticos. Finalmente, más del 8% (2/24) de los experimentos detectaron mejoras en tareas de programación.

Fijándose en los niveles de implicación donde se han detectado mejoras, existen dos extremos. *Cambio* es el más numeroso con un 37.5% (9/24) de los experimentos, mientras que *Presentación* es el opuesto con un 12.5% (3/24). El nivel de *Respuesta* está presente en el 20.8% (5/24) de los experimentos, y tanto el nivel de *Visión* como el de *Construcción* se han utilizado en el 27.2% (7/24) de los experimentos.

Tabla 1. Resumen de experimentos agrupados por niveles de implicación.

Experimento	Mejora educativa	Nivel de implicación	Contenidos textuales	Retroalimentación explícita	Tiempo añadido	Características avanzadas
Lawrence (1993) [17] Ch.7	Conocimiento	(V)	•			
Crosby y Stelovsky (1995) [4]	Conocimiento	(V) → NV	•			
Kann et al. (1997) [13]	Conocimiento	(V) → NV				
Keheo et al. (2001) [14]	Conocimiento	(V) → NV	•		•	
Kumar (2005) [15]	Conocimiento	(V) → NV	•			
Urqiza (2008) [22]	Conocimiento	(V) → NV	•			
Byrne et al. (1999) [3]	Conocimiento	(R) → NV	•			
Grisson et al. (2003) [6]	Conocimiento	(R) → NV	•			
Laakso (2005) [16]	Actitud	(R, V) → NV		•		
Hansen et al. (2000) estudios I-II-IV [7]	Conocimiento	(CH, R) → NV	•	•		Contenidos
Hansen et al. (2000) estudio V [7]	Conocimiento	(CH, R) → V	•	•		Contenidos
Lawrence (1993) Ch.6 [17]	Conocimiento	(CH) → V	•			
Lawrence et al. (1994) [18]	Conocimiento	(CH) → NV	•		•	
Ben-Bassat et al. (2003) [2]	Conocimiento	(CH)				
Moskal et al. (2004) [19]	Conocimiento	(CH) → NV			•	Interfaz del IDE
Ahoniemi y Lahrinen (2007) [1]	Conocimiento	(CH) → NV	•			
Cross (2007) [5] + Jain (2006) [12]	Programación	(CH) → NV				Interfaz de visualización
Stasko (1997) [21]	Actitud	(C, CH)				
Urqiza y Velazquez (2007) [23]	Act./Conoc.	(C) → V	•		•	
Urqiza (2008) [22]	Act. y Conoc.	(C) → NV	•			
Urqiza (2008) [22]	Actitud	(C) → V	•			
Hundhausen (2002) [9]	Actitud	(P, C)	•			
Hübsher-Yongier y Narayanan (2003) [8]	Conocimiento	(P, C) → V	•			
Hundhausen y Brown (2008) [10]	Programación	(P, C)	•			

No todos los experimentos son estudios comparativos, el 20.8% (5/24) exploran posibles mejoras dentro de niveles concretos. En cuanto a los estudios comparativos, el 73.7% (14/19) han comparado el uso de algún nivel de implicación con respecto a la ausencia total de animaciones. El resto, un 26.3% (5/19) hizo comparaciones con el nivel de *Visión*.

El uso de contenidos narrativos y explicaciones textuales está presente en el 75% (18/24) de los experimentos. Esto significa que deben ser un factor a tener en cuenta cuando se diseña una experiencia educativa con animaciones. Mientras que la retroalimentación explícita, el tiempo adicional de trabajo con visualizaciones, u otras características avanzadas —contenidos de alta calidad o interfaces avanzados— están presentes en más del 20% de los experimentos.

4.2 Recomendaciones para el diseño de experiencias educativas con animaciones

Al no ser este trabajo un meta-estudio, como el de Hundhausen et al. [11], no podemos dar evidencia formal y científica de las correlaciones existentes entre los niveles de implicación y las mejoras educativas detectadas. Sin embargo, todos estos experimentos dan evidencia empírica de usos educativos eficaces de los diferentes niveles de implicación, de forma que podemos extraer un conjunto de recomendaciones para cada nivel.

El simple hecho de **ver animaciones** puede ayudar a mejorar la adquisición de conocimientos, pero las animaciones deberán integrar explicaciones textuales y contenidos narrativos.

Cuando los estudiantes **responden a preguntas durante la reproducción** de una animación, de nuevo se les debería proporcionar contenidos narrativos y explicaciones textuales. La retroalimentación explícita también es importante. Aunque no está presente en dos de los experimentos, las preguntas planteadas en estos eran de tipo predictivo prácticamente inmediato, de forma que la respuesta correcta aparecía en los siguientes pasos de la animación.

En un ámbito de tareas más activas está la posibilidad de que los estudiantes **cambien los datos de entrada de los algoritmos**. En este caso, los contenidos narrativos y las explicaciones textuales parecen ser menos importantes (5/8). La razón podría ser que los investigadores se interesaron más en el trabajo cognitivo realizado por los estudiantes a la hora de seleccionar los datos de entrada, que en las explicaciones que se les podía facilitar sobre lo que pasa en la animación. Como esta tarea es de carácter explorativo, no se debería limitar el tiempo de trabajo de

forma estricta. Tampoco hay que olvidar las características avanzadas como los contenidos de alta calidad —animaciones de un mismo algoritmo en diferentes situaciones y con diferentes grados de abstracción [7]—, integración con el entorno de programación [19], o interfaces de usuario avanzadas para manipular las animaciones [5,12].

Como no podía ser de otra forma, cuando **los estudiantes construyen sus propias animaciones**, el interfaz de construcción toma gran importancia. Así, tanto proporcionar un interfaz diseñado específicamente para esta tarea como permitir a los estudiantes que elijan sus propias herramientas de construcción, son enfoques que han obtenido resultados positivos². Otro enfoque positivo ha sido el de animar a los estudiantes a generar sus propios contenidos narrativos o explicaciones textuales, en este caso las mejoras educativas se detectaron en términos de actitud de los estudiantes.

Finalmente, las tareas de **presentación de animaciones** han ido emparejadas con la construcción, por lo que el interfaz de nuevo es un punto clave.

4.3 Sugerencias para cambiar entre niveles de implicación

En este apartado analizamos los niveles de implicación que han sido mejorados por otros y de qué modo. Todos los experimentos han detectado mejoras respecto a los dos primeros niveles *Sin visión* y *Visión*.

Mejoras respecto al nivel *Sin visión*

El nivel *Sin visión* se caracteriza por la total ausencia de visualizaciones en la metodología pedagógica. Así, un cambio muy sencillo es trasladarse al siguiente nivel, *Visión*. Este cambio ha demostrado mejorar la adquisición de conocimiento de los estudiantes. Existen bastantes colecciones de animaciones de programas y algoritmos, pero si se quiere generar una propia, no hay que olvidar añadir contenidos narrativos y explicaciones textuales.

También es posible trasladarse al nivel de *Respuesta*, ya que también existen colecciones relacionadas con este nivel. En este caso se han detectado mejoras tanto de la actitud de los estudiantes como de adquisición de conocimientos. De nuevo, a la hora de generar sus propias animaciones hay que tener en cuenta la

² Ambas opciones representan un enfoque centrado en el estudiante, más que centrarse en proporcionar soluciones tecnológicamente avanzadas.

integración con contenidos narrativos, explicaciones textuales, así como retroalimentación explícita.

Se puede mejorar tanto la actitud de los estudiantes como su adquisición de conocimiento o su habilidad en tareas de programación si se traslada al nivel de *Cambio*. Probablemente, esto les llevará más tiempo de trabajo a los estudiantes, puesto que las actividades de este nivel se suelen enmarcar en un entorno de trabajo propio, más que laboratorios o clases. Si se usa este nivel junto con el de *Respuesta*, se deberían incluir contenidos narrativos, explicaciones textuales y retroalimentación explícita. Los otros experimentos realizados en este nivel utilizaron aspectos avanzados como: contenidos de alta calidad —lo que implica más trabajo por parte del profesor—, integración con el entorno de programación e interfaces avanzadas con las visualizaciones y con el entorno —lo que implica un mayor esfuerzo de desarrollo si se quiere producir su propio sistema—.

Finalmente, si se traslada al nivel de *Construcción*, se puede mejorar la actitud de los estudiantes y su adquisición de conocimientos. El proceso de construcción debe ser sencillo, y debería incluir la integración de contenidos narrativos y explicaciones textuales.

Mejoras respecto al nivel *Visión*

En este nivel, la interacción con las animaciones suele ser escasa. Así que, un cambio simple es trasladarse al nivel de *Cambio*, donde se han detectado mejoras en la adquisición de conocimiento. Las características de las animaciones que produjeron estas mejoras eran la integración de contenidos narrativos y explicaciones textuales, así como los contenidos de alta calidad.

El traslado al nivel de *Construcción* puede mejorar la actitud de los estudiantes, y como efecto secundario, la adquisición de conocimiento. Al igual que antes, es necesario que el proceso de construcción sea sencillo, permitiendo integrar contenidos narrativos y explicaciones textuales. Se puede utilizar junto con el nivel de *Presentación*, mejoran la adquisición de conocimientos, pero los estudiantes deberían poder elegir sus propias herramientas de construcción de animaciones.

5 Conclusiones y trabajos futuros

Esto no es un meta estudio formal, puesto que no hemos incluido los experimentos que no han conseguido detectar alguna mejora, por lo tanto no podemos dar evidencia científica de que las características estudiadas sean factores decisivos

para conseguir que las animaciones sean herramientas pedagógicas eficaces. Sin embargo, podemos dar un conjunto de recomendaciones. Hemos constatado la existencia de ciertas características en muchos de los experimentos considerados: la integración de contenidos narrativos y explicaciones textuales, retroalimentación sobre las respuestas de los estudiantes a preguntas realizadas y un enfoque centrado en el estudiante a la hora de diseñar herramientas de construcción de animaciones. Finalmente, hemos identificado posibles traslados entre diferentes niveles de implicación y los efectos en el ámbito educativo.

El trabajo futuro obviamente incluye la consideración de experimentos sin resultados positivos, y así poder detectar correlaciones entre los niveles de implicación y las mejoras educativas con mayor validez científica. Aunque tampoco olvidamos la posibilidad de incluir otras características.

6 Bibliografía

1. T. Ahoniemi y E. Lahtinen. Visualizations in preparing for programming exercise sessions. *Electronic Notes in Theoretical Computer Science*, 178:137-144, July 2007.
2. R. Ben-Bassat, M. Ben-Ari, y P.A. Uronen. The jeliot 2000 program animation system. *Computers & Education*, 40(1):1-15, January 2003.
3. M.D. Byrne, R. Catrambone, y J.T. Stasko. Evaluating animations as student aids in learning computer algorithms. *Computers & Education*, 33:253-278, 1999.
4. M.E. Crosby y J. Stelovsky. From multimedia instruction to multimedia evaluation. *Journal of Educational Multimedia y Hypermedia*, 4:147-162, 1995.
5. J.H. Cross, T.D. Hendrix, J. Jain y L.A. Barowski. Dynamic object viewers for data structures. In *SIGCSE '07: Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, pages 4-8, New York, NY, USA, 2007. ACM Press.
6. S. Grissom, M.F. McNally, y T.L. Naps. Algorithm visualization in CS education: comparing levels of student engagement. In *Proceedings of the 2003 ACM Symposium on Software Visualization*, pages 87-94, New York, NY, USA, 2003. ACM Press.
7. S.R. Hansen, N.H. Narayanan, y D. Schrimpscher. Helping learners visualize y comprehend algorithms. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 2(1), April 2000. <http://imej.wfu.edu/articles/2000/1/02/>, 2008.
8. T. Hübscher-Younger y N.H. Narayanan. Dancing hamsters y marble statues: characterizing student visualizations of algorithms. In *SoftVis '03: Proceedings of the 2003 ACM Symposium on Software Visualization*, pages 95-104, New York, NY, USA, 2003. ACM Press.

9. C.D. Hundhausen. Integrating algorithm visualization technology into an undergraduate algorithms course: ethnographic studies of a social constructivist approach. *Computers & Education*, 39(3):237-260, 2002.
10. C.D. Hundhausen y J.L. Brown. Designing, visualizing, y discussing algorithms within a cs 1 studio experience: An empirical study. *Computers & Education*, 50(1):301-326, 2008.
11. C.D. Hundhausen, S.A. Douglas, y J.T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages y Computing*, 13(3):259-290, 2002.
12. J. Jain, J.H. Cross, T.D. Hendrix y L.A. Barowski. Experimental evaluation of animated-verifying object viewers for java. In *SoftVis'06: Proceedings of the 2006 ACM symposium on Software visualization*, pages 27-36, New York, NY, USA, 2006. ACM.
13. C. Kann, R.W. Lindeman, y R. Heller. Integrating algorithm animation into a learning environment. *Computers & Education*, 28(4):223-228, 1997.
14. C. Kehoe, J.T. Stasko, y A. Taylor. Rethinking the evaluation of algorithm animations as learning aids: An observational study. *International Journal of Human-Computer Studies*, 54(2):265-284, 2001.
15. A.N. Kumar. Results from the evaluation of the effectiveness of an online tutor on expression evaluation. In *SIGCSE '05: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, pages 216-220, New York, NY, USA, 2005. ACM Press.
16. M-J. Laakso, T. Salakoski, L. Gryell, X. Qiu, A. Korhonen y L. Malmi. Multi-perspective study of novice learners adopting the visual algorithm simulation exercise system TRAKLA2. *Informatics in Education*, 4(1):49-68, 2005.
17. A.W. Lawrence. Empirical studies of the value of algorithm animation in algorithm understying. PhD thesis, Dep. of Computer Science, Georgia Institute of Technology, 1993.
18. A.W. Lawrence, A.M. Badre, y J.T. Stasko. Empirically evaluating the use of animations to teach algorithms. In *IEEE Symposium on Visual Languages*, 1994. Proceedings, pages 48-54. Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.
19. B. Moskal, D. Lurie, y S. Cooper. Evaluating the effectiveness of a new instructional approach. In *Proceedings of the Technical Symposium on Computer Science Education, SIGCSE'04*, pages 75-79, New York, NY, USA, 2004. ACM Press.
20. T.L. Naps, S. Cooper, B. Koldehofe, C. Leska, G. Rößling, W. Dann, A. Korhonen, L. Malmi, J. Rantakokko, R.J. Ross, J. Yerson, R. Fleischer, M. Kuittinen y M. McNally. Iticse 2002 working group report: Exploring the role of visualization y engagement in computer science education. *ACM SIGCSE Bulletin*, Working group reports from ITiCSE on Innovation y technology in computer science education, 35(2):131-152, june 2002.

Eficacia Pedagógica de los Niveles de Implicación de los Estudiantes con las Animaciones de Programas y Algoritmos 15

21. J.T. Stasko. Using student-built algorithm animations as learning aids. In SIGCSE '97: Proceedings of the twenty-eighth SIGCSE Technical Symposium on Computer Science Education, pages 25-29, New York, NY, USA, 1997. ACM Press.
22. J. Urquiza-Fuentes. Generación Semiautomática de Animaciones de Programas Funcionales con Fines Educativos. Tesis doctoral, Dep. de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos, 2008.
23. J. Urquiza-Fuentes y J.Á. Velázquez-Iturbide. An evaluation of the effortless approach to build algorithm animations with winhipe. *Electronic Notes in Theoretical Computer Science*, 178:3-13, July 2007.