Toward the effective use of educational program animations: The roles of student's engagement and topic complexity

Jaime Urquiza-Fuentes, J. Ángel Velázquez-Iturbide

Computers & Education, 2013, 67, 178-192.

DOI: http://dx.doi.org/10.1016/j.compedu.2013.02.013

Towards the Effective Use of Educational Program Animations: the Roles of Student's Engagement and Topic Complexity

Jaime Urquiza-Fuentes[a,*], J. Ángel Velázquez-Iturbide[a]

[a]*Departamento de Lenguajes y Sistemas Informáticos I , Universidad Rey Juan Carlos,C/Tulipán s/n,28933 Móstoles, Madrid, Spain*

Abstract

Programming is one of the most complex subjects in computer science degrees. Program visualization is one of the approaches adopted to make programming concepts more accessible to students. In this work we study the educational impact of an active and highly engaging approach, namely the construction of program animations by students. We systematically compared this approach with two instructional scenarios, based on viewing animations and on the traditional instruction without systematic use of animations. A general conclusion of this work is that animations actually improve learning in terms of some educational aspects: short-term and long-term knowledge acquisition, and drop-out rates. Short-term improvements depend on the complexity level of the topic: while there is no impact for simple topics, there is a learning improvement in complex topics using the viewing and constructing approaches, and there is a learning improvement for highly complex topics using the viewing approach. In the long-term, drop-out rates were significantly decreased for students involved in the two most engaging approaches. In addition, both animation viewing and animation construction improved students passing-rate in the term exam. Nevertheless, we were unable to prove in the long term that students involved in construction tasks yielded higher grades than those involved in viewing tasks.

Keywords: evaluation of CAL systems, interactive learning environments,

*Corresponding author

Email addresses: jaime.urquiza@urjc.es (Jaime Urquiza-Fuentes), angel.velazquez@urjc.es (J. Ángel Velázquez-Iturbide)

multimedia/hypermedia systems, programming and programming languages, simulations

---

1. Introduction

Learning to program is one of the most difficult challenges in computer science education. External representations of programs' behavior, i.e. program and algorithm animations (animations for the rest of the paper), are one approach to overcome its difficulties. Animations increase student motivation and can be designed to serve several learning purposes (Ainsworth, 1999). In fact, animations have been used in educational contexts since the seventies (Baecker, 1998).

Literature on educational software visualization (Anderson and Naps, 2001; Ben-Ari, 2001; Kehoe et al., 2001; Saraiya et al., 2004; Stasko et al., 1998) reports on successful experiments regarding the effectiveness of animations in education. However, not all the existing experiments support this intuition. In their meta-analysis of experiments, Hundhausen et al. (2002) conclude that the more active the educational use, the better the learning outcomes.

Based on Hundhausen et al.'s results, a working group on visualization (Naps et al., 2002) developed the *Engagement Taxonomy*, a proposal to classify different educational uses of animations. They suggested that the higher the students' engagement with animations, the better the learning outcomes. The engagement taxonomy proposed six forms of students' engagement with animations: no-viewing, viewing, responding, changing, constructing and presenting.

A second critique to animations is the limited success they have in the community of CS educators. A survey conducted by the aforementioned working group (Naps et al., 2002) reported that a major factor instructors argued for not adopting visualizations was the great effort visualization software demanded to them. Consequently, the concept of "effortlessness" was coined, applied to a software visualization system that hardly demanded instructors an effort for adopting and adapting it to a course.

The concept of effortlessness also applies to students, for the case they will use the system to construct new animations. This is an important feature as, according to Mayer (2005), meaningful learning requires learner's cognitive activity rather than learner's behavioral activity. Note that this means that some behavioral activities could not promote cognitive activity (Mayer, 2005). Stated in these terms, an effortless visualization system would allow constructing new visualizations with little behavioral activity, thus leaving room to the cognitive activity associated to the active learning task.

In an effort to explore the feasibility of developing an effortless visualization system, we constructed the WinHIPE system (Pareja-Flores et al., 2007). It is an integrated development environment, extended with visualization facilities for the functional programming language Hope (Patterson, 2010). We found significant evidence that students considered PACT easy to use independently of their previous experience with it and that most students used the system even when they were not urged to do so (Velázquez-Iturbide et al., 2008).

In this article we go one step forward in our research by inquiring whether the use of an effortless visualization system for construction tasks, as ours, may yield better educational outcomes. Our research was organized with three groups: the experimental group that constructs animations, and two control groups. One control group did not use animations at all, and the second control group had the most common form of engagement, i.e. viewing. Thus, we used the students' engagement level with animations as the independent variable of our research.

Viewing can take different forms, and evaluations have yielded opposite results. We wanted to have the viewing control group in the best of the circumstances. Other authors have distinguished between active and passive viewing, depending on whether the user has control over the animation (Lauer, 2008; Myller et al., 2009). Obviously, active viewing is the most engaging form of viewing. Moreover, the animations used had a narrative component. According to the Dual Coding Theory (Paivio, 1983) and the principles proposed by Mayer (2005) for multimedia learning design, integrating narratives into animations may help the students to better understand the concepts explained within these animations.

Our research question may be posed in general terms, but we may also inquire into more specific factors. In particular, one of the features of deep learning is retention of knowledge in the long term. Can this feature be promoted by an engaging activity, as visualization construction, supported by an effortless visualization system?

Short-term and long-term evaluations have different features, therefore we may use them to inquire into different, more specific factors. In particular, short-term evaluations must be held when some specific topic of a course is being taught. Therefore, we may inquire whether construction tasks enhance learning for any degree of difficulty of the topic of study.

Knowledge acquisition is a general concept, but educational theories allow speaking about different degrees of student achievement. For instance, Bloom's taxonomy (Bloom et al., 1959) states six levels, from lower to higher achievement: knowledge, comprehension, application, analysis, synthesis and evaluation. Consequently, we may also measure the effect of construction tasks on knowledge acquisition at different levels of Bloom's taxonomy.

Another consequence of effortlessness is that the users may be more pleased using the system, given that they may focus on the target task. Satisfaction with a system in the short term is mostly related to usability, while satisfaction in the long term is mostly related to user experience. We consider that this second factor is more interesting in our context. Therefore, we have also measured learners' satisfaction using our WinHIPE system in the long term.

1.1. Research questions

This investigation addresses several research questions. The first research question regarding program animation that is typically posed is: *does the use of program animations improve students' learning outcomes?* There is an extensive literature supporting the use of graphical representations to improve learning outcomes. From a general point of view, Larkin and Simon (1987) explain how diagrammatic representations could improve understanding with respect to sentential representations. And focusing on learning, Ainsworth (1999) shows how external graphical representations can be effective learning tools, while Mayer (2005) describes the features that multimedia presentations should have to improve learning. Furthermore, several reviews of the program and algorithm animation field (Hundhausen, 2002; Urquiza-Fuentes and Velázquez-Iturbide, 2009) show that animations can improve students' learning outcomes. As a consequence, we could expect to have a positive answer to this question.

One of the Cognitive Constructivism principles states that learning is an active process. Learners actively construct new understandings by becoming actively engaged with their environment, e.g. Flavell (1989). Other authors have highlighted the importance of interactivity in external representations (Scaife and Rogers, 1996). In the field of algorithm animation, Hundhausen et al.'s (2002) review concludes that what students do with animations is more important than what animations show to the students. Therefore, a second question arises regarding the engagement taxonomy: *does an increase in students' engagement with animations lead to improvements in their learning outcomes?*

A third research question focuses on: *under what circumstances the improvements in the learning outcomes are detected?* This guided the election of the dependent variables of our study. As it was argued above, the short-term effects of the treatments were measured not only in terms of knowledge acquisition, but also in terms of different degrees of student achievement and different levels of topic complexity. Long-term effects were measured in terms of global knowledge acquisition, drop-out rates and learners' satisfaction.

1.2. Contributions

The main contribution of the work is to clarify the educational use of the viewing and constructing tasks of program animation. An increase in the students' engagement level leads to a decrease in drop-out rates. But the effect on students' knowledge acquisition is not so clear. This effect is conditioned by the length of time spent with animations and the complexity of the topics dealt in the animation. Neither of these uses of animations (viewing or constructing) improved knowledge acquisition for simple topics. However, topics of medium complexity are suitable for both uses. Finally, only the viewing tasks produced improvements in knowledge acquisition for highly complex topics. With respect to the level of student achievement, all the improvements in knowledge acquisition were detected at the analysis and synthesis levels of Bloom's taxonomy.

The rest of the article is structured as follows. First we survey related research in section 2. In section 3 we describe the experimental design and method of the study. We detail the results of the study in sections 4 and 5. In section 6 we discuss the results of the study. Finally, we draw our conclusions and future work in section 7.

2. Related Work

As explained in the introduction, we assume the *Engagement Taxonomy* (Naps et al., 2002) as a framework to classify students engagement with visualizations. The taxonomy identifies six levels of engagement: *No-viewing*, *Viewing*, *Responding*, *Changing*, *Constructing* and *Presenting*.

The *No-viewing* level represents the traditional scenario with no use of animations. In the *Viewing* level, students can play animations, changing their direction, pace or abstraction level. The *Responding* level consists in asking questions to students during the animation. In the *Changing* level, students are able to explore the behavior of the algorithm by changing the input data. The *Constructing* level requires from students to construct explanatory animations of the algorithm behavior. Finally, in the *Presenting* level students have to present an animation to an audience.

The taxonomy does not establish a strict hierarchy for the engagement levels, but some levels seem to demand more engagement from the student than others. In this sense, it is common to give the same engagement order for the levels as they were introduced above, i.e. being *No-viewing* the least engaging level and *Presenting* the most engaging one.

Our work is focused on the *Construction* level, comparing it with the *Viewing* level and the *No-viewing* level. There are important contributions regarding other engagement levels as *Responding* (Grissom et al., 2003) or *Changing* (Ben-Bassat et al., 2003), but in this section we only survey related experiments that evaluate the educational impact of tasks for animation viewing or constructing.

2.1. Review of experiments

We have found eleven experiments regarding viewing or constructing activities of program animation. Table 1 summarizes the main features and results of these experiments. Two important lessons can be learned from these experiments:

1. It could be obvious, but the construction utilities that the user has available play an important role in construction tasks. Most of the studies related to construction allow students to choose the construction utilities (Hübscher-Younger and Narayanan, 2003; Hundhausen and Douglas, 2000; Hundhausen, 2002) or provide them with carefully designed interfaces for this task (Hundhausen and Brown, 2008; Urquiza-Fuentes and Velázquez-Iturbide, 2007).
2. The use of narratives and textual contents integrated with animations improves their effectiveness. Notice that they are present in most successful studies, either related to viewing tasks (Lawrence, 1993; Kann et al., 1997; Kehoe et al., 2001; Kumar, 2005) or to constructing tasks (Hübscher-Younger and Narayanan, 2003; Hundhausen and Douglas, 2000; Hundhausen, 2002; Hundhausen and Brown, 2008; Urquiza-Fuentes and Velázquez-Iturbide, 2007). This result is also supported by Mayer's (2005) contributions.

Table 1: Summary of experiments. The "Engagement & Results" column represents the engagement levels involved in the study and its results; NV,V,Ch,C and P stand for the No-viewing, Viewing, Changing, Constructing and Presenting levels, respectively. When two different levels were jointly used they appear with an & symbol. When the study compares two levels, "X=Y" denotes that no significant differences were found, and "X>Y" denotes that significant differences were found, with X obtaining better results than Y. The results of each experiment also are encoded regarding the dependent variables used, namely student's attitude (SA), knowledge acquisition (KA) or programming performance (PP).

Experiment	Engagement & Results	Evaluation	Taxonomy
Stasko (1997)	C (SA)	Long term	–
Hundhausen and Douglas (2000)	C=Ch (PP)	Short term	–
Hundhausen (2002)	C&P (SA)	Short term	–
Hübscher-Younger and Narayanan(2003)	C>V (KA)	Short term	Other
Urquiza-Fuentes and Velázquez-Iturbide(2007)	C>V (SA/KA)	Short term	Bloom's
Hundhausen and Brown (2008)	C&P (PP)	Short term	–
Lawrence (1993)	V (KA)	Short term	Other
Crosby and Stelovsky (1995)	V>NV (KA)	Short term	–
Kann et al. (1997)	V>NV (KA)	Short term	Other
Kehoe et al. (2001)	V>NV (KA)	Short term	Bloom's
Kumar (2005)	V>NV (KA)	Short term	–

We have only found two comparative studies involving animation construction tasks. Both studies showed positive results and reinforce the notion that motivation plays a role in learning. On the one hand, Urquiza-Fuentes and Velázquez-Iturbide's (2007) study showed how students who constructed animations kept working with them more time than students who just viewed animations. As a consequence, the former group got better results. On the other hand, participants from Hübscher-Younger and Narayanan's (2003) study voluntarily chose between just viewing animations or both viewing and constructing animations. The students who chose the latter option had to make a bigger effort (which means that they were probably more motivated) and got better results.

Since all the studies but one are short-term evaluations, the continuous use of program animations within a course lacks of any systematic research.

Finally, the measurement of the learning outcomes does not seem to follow a systematic approach. Only two studies characterize learning outcomes based on Bloom's taxonomy. Three studies use other classifications, such as differentiating between theoretical and practical exercises, or between factual and procedural knowledge. The rest of the studies do not use any taxonomy or classification scheme.

We consider that addressing these issues will contribute to improve our knowledge about the educational impact of students' engagement with program animations. Consequently, our study compares three engagement levels –no-viewing, viewing and constructing– under their best circumstances, considering short-term and long-term effects, and measuring learning outcomes in terms of the Bloom's taxonomy. The details of the study are explained in the next section.

3. Experimental Design and Method

In this section we describe the experimental design and method. We first explain the design of the experiment (hypotheses, independent and dependent variables). Then we describe the participants and the course where the evaluation was conducted. Next, we characterize the animations used for the two uses of animation under study: viewing and constructing. Finally, we detail the tasks and the protocol followed in the study.

3.1. Hypotheses

We derive three hypotheses from our three research questions. From the first research question "*does the use of program animations improve students' learning outcomes?*" we derive two hypotheses:

H1 The use of animation viewing activities outperforms the no-viewing approach.

H2 The use of animation constructing activities outperforms the no-viewing approach.

The second research question "*does an increase in students' engagement with animations lead to improvements in their learning outcomes?*" allow us to derive the third hypothesis:

H3 The use of animation constructing activities outperforms the use of animation viewing activities.

Table 2: Summary of hypotheses.

	Long-term Knowledge acq.	**Short-term** Knowledge acq.	**Drop-out** rate
Viewing better than no-viewing	H1a	H1b	H1c
Constructing better than no-viewing	H2a	H2b	H2c
Constructing better than viewing	H3a	H3b	H3c

According to the third research question "*under what circumstances the improvements in the learning outcomes are detected?*", each of these hypotheses is subdivided into three sub-hypotheses regarding: (a) students' short-term knowledge acquisition, (b) students' long-term knowledge acquisition and (c) drop-out rates. Therefore, our study is based on nine sub-hypotheses, resulting from combining comparisons of educational uses and measured effects (see Table 2).

3.2. Design of the Experiment

The long-term evaluation included three short-term evaluations. In each short-term evaluation, the treatments were applied during a short period of time, and the effects were measured immediately after the treatment. In the long-term evaluation, we considered the continuous use of the treatments during all the course and the effects were measured at the end.

The independent variable was the engagement level in terms of Naps et al.'s (2002) taxonomy. The treatment group was CG (constructing engagement level), while VG (viewing engagement level) and TG (the traditional, no-viewing engagement level) were used as control groups.

In both the short-term and the long-term evaluations, the dependent variable was knowledge acquisition. In the long-term evaluation, we also investigated drop-out rates and learners' satisfaction.

3.2.1. Dependent variables and measurement instruments

Regarding knowledge acquisition, the measurement instrument used in the short-term evaluations was a set of three knowledge tests about three different topics of the course: infix operators, user defined data types, and recursive data types. These topics represent different levels of complexity: low, medium and high. In addition, the questions in these knowledge tests were classified according to the following levels of Bloom's taxonomy: knowledge, understanding, application, analysis and synthesis. We followed Horn's classification algorithm (De Landsheere, 1997) to map the questions in knowledge tests into levels of Bloom's taxonomy. An example of knowledge test can be found in Appendix A. Thus, we consider the dependent variable knowledge acquisition from three points of view:

Global point of view: all the evaluated levels of Bloom's taxonomy and all the topics were combined into one value per student.

Bloom-level point of view: for each level of Bloom's taxonomy, all the topics were combined into one value. Therefore we obtained five values per student.

Topic point of view: for each topic, all the levels of Bloom's taxonomy evaluated were combined into one value. Therefore we obtained three values per student.

The measurement instrument used to measure knowledge acquisition in the long-term evaluation was the term exam of the subject. Consult the term exam in Appendix B.

In the long-term evaluation, we also measured drop-out rates and learners' satisfaction. We considered that participants drop the subject out if they did not take the term exam. Our rationale was that if students do not drop the subject out, we may consider that they have a positive attitude towards the subject. This is especially important in a course on such a complex subject matter as foundations of programming languages offered in the first year of computing degrees.

Learners' satisfaction was measured with an opinion questionnaire about the educational use of animations. The students gave their opinion about four statements using a Likert scale of five values (totally agree, partially agree, no opinion, partially disagree and totally disagree). The statements were: (1) animations have helped me in understanding the concepts, (2) animations are easy to construct, (3) animations are easy to use, and (4) animations are useful. Note that statement (2) was only answered by students from CG while statement (3) was only answered by students from VG. Students' opinion gives complementary data because we could expect that the more comfortable students feel with program animations, the greater possibilities they have to obtain educational benefits.

Table 3: Summary of dependent variables, evaluation terms and measurement instruments.

	Short-term	Long-term
Knowledge acquisition	Knowledge tests: Global, Bloom-level & Topic	Term exam
Drop-out rate	–	Term exam
Learner's satisfaction	–	Opinion questionnaire

Table 3 summarizes all dependent variables used in this study, showing measurement instruments and long/short term evaluation.

3.3. Participants

All the participants were computing majors enrolled in a course on foundations of programming languages, described below. A total of 132 students participated in the evaluation. Three different enrolment groups were used, so the mapping between students and groups was independent from the experiment. Treatment/control conditions were randomly assigned to the three groups (CG, VG and TG).

The participation was mostly voluntary, so the number of students slightly varied among groups: TG (n=42, out of 149), VG (n=50, out of 120) and CG (n=40, out of 119). Only four participants were female, one in TG, two in VG and 1 in CG. The students' average age was 19.1. As the amount of time dedicated to voluntary tasks was significant, participants were rewarded with a small grade incentive: if they passed the term exam, they would have their grades 0.25 points higher (out of 10.0). The term exam was compulsory for all the students (participants or not) who wanted to pass the course. We asked the voluntary students about their knowledge on the functional paradigm and we discarded students with previous knowledge, as well as students who had been enrolled in the course in previous years.

3.4. The Educational Context

The context of this evaluation is the mandatory course Foundations of Programming Languages, offered to freshmen in two computing degrees –management computing and computing systems– at the Rey Juan Carlos University. The evaluation was conducted in the second half of the course, which is scheduled as an introduction to the functional programming paradigm. The course has a length of fifteen weeks, with eight weeks devoted to the functional paradigm with the following structure.

1. The functional paradigm.
2. Basic features of the functional language Hope (Patterson, 2010).
3. Recursion.
4. Prefix and infix operators.
5. User-defined data types.
6. Local definitions.
7. Recursive data types.
8. Polymorphic data types.

Topics 1, 2 and 3 were taught during the first two weeks. Topics 4, 5, 6 and 7 were taught during the following five weeks. Finally, topic 8 and a general review of the course were taught in the last week of the course.

3.5. Educational Materials

In this evaluation we investigate the effects of *viewing* and *constructing* program animations as educational tools. Both engagement levels are instantiated with the effortless approach to animation construction of the WinHIPE functional programming environment (Pareja-Flores et al., 2007); a detailed description of the approach can be found elsewhere (Velázquez-Iturbide et al., 2008). We first outline the structure of the animations, which were used for the viewing engagement level, and then describe the approach to animation construction.

3.5.1. Format and Contents of Educational Animations of WinHIPE

The WinHIPE environment allows generating proprietary animations, as well as web-based animations. We used the latter for our experiment. Web animations consist of four components: a problem statement, an explanation of a solution to the problem (typically, an algorithm), source code that implements it, and the program animation itself. The pace and direction of animations can be controlled with a typical VCR interface. Program animations can simultaneously display either one or two consecutive steps of the program execution. Fig. 1 shows an example of an animation regarding simple recursive data types. It illustrates the computation of the sum of a list of integers with all of its contents visible and showing two consecutive steps of the program execution.

Web animations may be structured according to several layout formats, but these details are not relevant for the experiment, and can be found elsewhere (Urquiza-Fuentes and Velázquez-Iturbide, 2005).

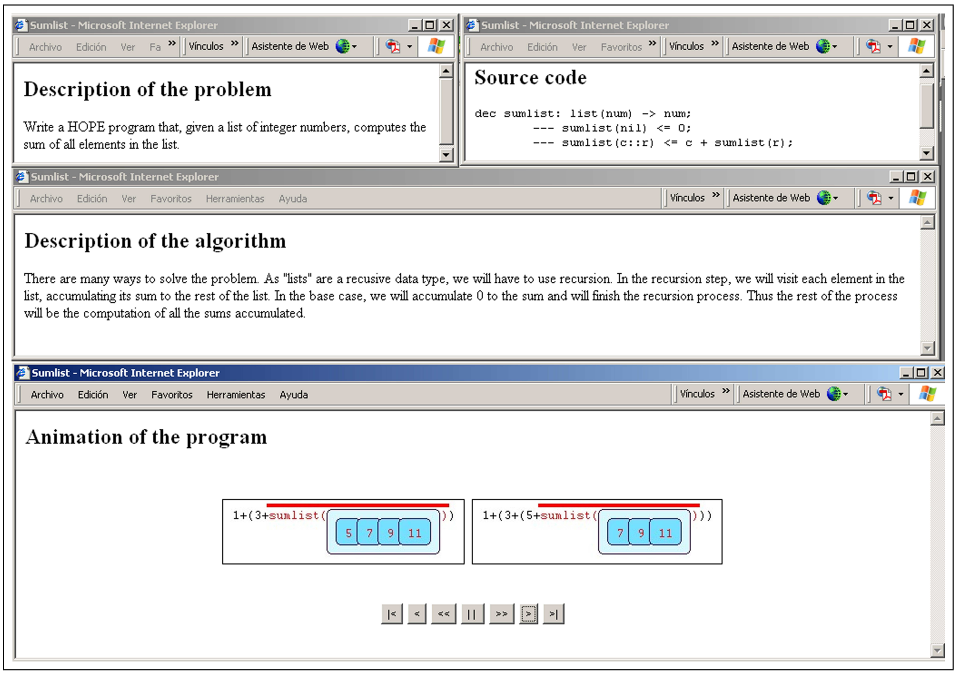

Figure 1: An educational animation about the sum of a list of integer values

3.5.2. Constructing Educational Animations with WinHIPE

The design and use of educational animations in WinHIPE are based on the metaphor of animations as office documents. An important feature of office applications is their facility to produce documents with a user-friendly graphical user interface, without the need to learn any language. Furthermore, office documents not only can be constructed, but they can also be stored, retrieved and modified. Following such a metaphor, we considered animations delivered by our programming environment as *electronic documents* that should be easy to construct and modify.

Notice that WinHIPE is a programming environment for functional programming. The execution model of the functional paradigm is term rewriting. Any expression built from declarations of the program can be evaluated. Evaluation is a term rewriting process, where an initial expression is rewritten into equivalent expressions, according to the program, until no rewriting can be done. The resulting expression is the value of the original expression. WinHIPE allows controlling and inspecting an evaluation with typical debugging facilities. Therefore, the user can access the whole trace of her/his

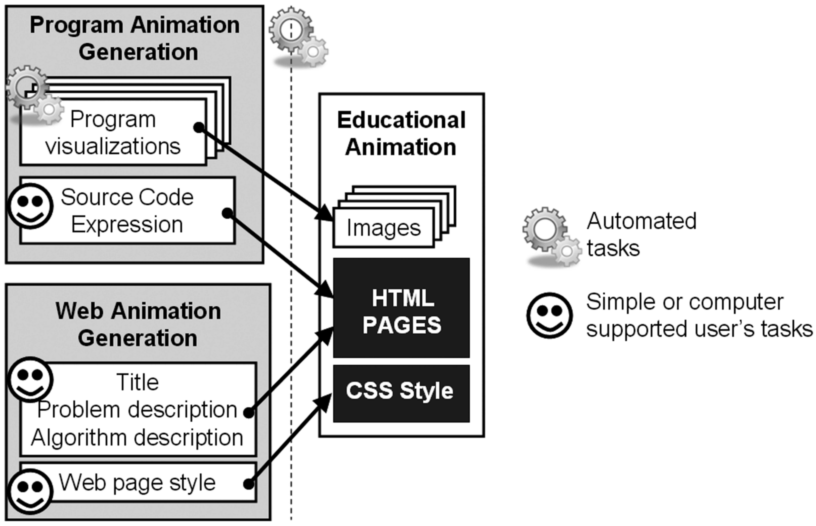

Figure 2: The construction process of educational animations

program execution, i.e. a complete sequence of expressions, from the original expression to its value.

The construction of our educational animations is divided into two phases: generation of the program animation (i.e. the graphical representations of the execution of a program) and generation of the educational animation (i.e. the web-based materials). See Fig. 2 for a schematic view of the process. We elaborate these phases a bit more:

1. Generation of the program animation, i.e. the sequence of visualizations. In turn, it consists of two subphases:
   (a) An expression is entered and evaluated. During its evaluation, each intermediate expression is displayed in a mixed format of pretty-printed text and built-in graphical representations for lists and binary trees. In this phase, the user must only type the expression to evaluate and select with the mouse the rewriting operations to perform.
   (b) The user selects the visualizations that will form the target animation. By default, all of the visualizations are selected, but the user may exclude irrelevant ones with mouse clicks.
2. Generation of the web animation. The user must provide textual contents (i.e. the problem statement and an explanation of the algorithm)

and specify look contents (i.e. its appearance on the web). This final step generates textual contents (as HTML code) and look information (as CSS code), and integrates them with the source code and the visualizations that will form the program animation.

Notice that user interaction is limited to controlling the expression evaluation, (optionally) selecting the relevant visualizations, writing an explanation of the problem and the algorithm, and specifying the web look of the animation. Any additional processing is automatic.

3.6. Tasks

Each of the three groups performed different types of tasks but using the same amount of time and similar problems, all of them selected from the course textbook (Velázquez-Iturbide et al., 2005).

Students assigned to the TG control group worked on solving problems proposed by the teacher. They were allowed to use the WinHIPE environment without its visualization facilities. Students had to solve problems by themselves but the teacher could help them on demand.

The task for students assigned to the VG control group was viewing animations previously constructed with WinHIPE by the teacher (see subsection 3.5.1). For each topic, students had available a set of animations at a web page. They had to view the animations ensuring that they understood them. The teacher had little or no interaction with students during the sessions of this group.

The task for students assigned to the treatment group (CG) was constructing animations with WinHIPE (see subsection 3.5.2). They had available at the web a description of the problem and source code. Therefore, the construction task consisted in studying the program and building an educational animation. On the one hand, they had to write an explanation of the solution implemented in the source code. On the other hand, they had to select an appropriate input data set, execute the program and generate the animation using the visualization facilities provided by WinHIPE.

3.7. Protocol

The course was fifteen weeks long. At the beginning of the course, students were informed about the evaluation. During the first, second and third topics (see subsection 3.4), students became familiar with WinHIPE. Students in VG or CG were also trained in the facilities provided by the

		In-class sessions	
		Theoretical	Laboratory
Groups	TG	Lecture + examples	Exercises + knowledge test
	VG	Lecture + examples+ **animations**	**View animations** + knowledge test
	CG		**Construct animations** + knowledge test

Table 4: Protocol followed in theoretical and lab sessions of the three groups

environment for the tasks they had to complete, i.e. to view or to construct animations.

The treatments were applied during five weeks in the sessions corresponding to topics 4, 5 and 7. Topics 6 and 8 were taught to the three groups with a traditional methodology. The last week of the course was dedicated to a general review of the subject. The term exam was held two weeks later.

Topics 4, 5 and 7 followed an evaluative design. This design consisted of two parts: theoretical sessions (2 hours per session), and lab sessions (2 hours per session) where a number of tasks and a knowledge test were completed by the participants (see Table 4). TG followed a traditional learning approach based on lectures and examples in the theoretical sessions and exercises in the lab sessions. VG and CG were taught similar contents, but lectures and examples were enhanced with animations. Their lab sessions were different: students from VG performed viewing tasks while students from CG performed construction tasks, as described in the previous subsection.

In summary, the students participated in three short-term evaluation sessions, completing three groups of tasks and three knowledge tests. Later, students were asked to fill in an opinion questionnaire. In addition, students had the term exam at the end of the course.

4. Short-Term Results

In this section we detail the results of short-term measurements of the dependent variable knowledge acquisition. As explained above (see subsection

Table 5: Average grades for each group and ANOVA analysis of global knowledge acquisition.

CG	VG	TG	ANOVA analysis
.6075	.6068	.5140	$F(2, 120) = 3.274, p = \mathbf{.041}$

Table 6: A detailed comparative study of global knowledge acquisition. Cells show the level of significance and differences (% of improvement) between pairs of groups.

Groups	t-Student analysis	Differences
CG vs. VG	$t(78.450) = .018, p = .986$	-
CG vs. TG	$t(74.360) = 2.311, p = .024$	-
VG vs. TG	$t(85) = 2.102, p = .038$	-

3.2.1), these measurements were collected from three different knowledge tests. We compared knowledge acquisition of the three groups from three different points of view: global, Bloom's level and topic.

For each point of view, the data analysis comprises a comparative study of the three groups and a detailed study of three pair-wise comparisons. On the one hand, the comparison among the three groups is performed with ANOVA test if data have a normal distribution; otherwise Kruskal-Wallis test is used. On the other hand, pair-wise tests are performed with t-Student if data have a normal distribution; otherwise Mann-Whitney test is used. We take into account that a multiple-comparisons analysis like this can generate type 1 errors. Therefore, to avoid these errors, we use Bonferroni adjustment on the results of pair-wise tests. The significance cut-off of pair-wise tests is placed on .016, i.e. the standard cut-off (.05) divided by the number of possible comparisons (3).

4.1. Global knowledge acquisition

This point of view offers a global view of students' learning outcomes. The ANOVA analysis (see table 5) shows that significant differences exist among the three groups. CG and VG seem to obtain better results than TG, but there are not significant differences between individual groups (see table 6). Therefore, we cannot confirm hypotheses H1a, H2a or H3a from a global point of view. A deeper analysis is performed in the next subsections.

Table 7: Average grades for each group and level of Bloom's taxonomy, and ANOVA or Kruskal-Wallis analysis of knowledge acquisition.

	CG	VG	TG	ANOVA or Kruskal-Wallis analysis
Knowledge	.6019	.5348	.5837	$F(2, 120) = 1.084, p = .342$
Understanding	.6525	.6794	.6316	$F(2, 120) = .561, p = .572$
Application	.6623	.6988	.5972	$H(2) = 2.899, p = .235$
Analysis	.6675	.6009	.4487	$H(2) = 19.175$, $p =$ **.01**
Synthesis	.4532	.5200	.3087	$F(2, 120) = 7.268, p =$ **.001**

Table 8: A detailed comparative study of Bloom-level knowledge acquisition regarding the *Analysis* level. Cells show the level of significance and the differences (% of improvement) between pairs of groups.

Groups	t-Student or Mann-Whitney U analysis	Differences
CG vs. VG	$t(78.450) = .018, p = .986$	-
CG vs. TG	$U = 318.0$, $p <$ **.001**	$CG > TG, 21.88\%$
VG vs. TG	$U = 596.5$, $p =$ **.003**	$VG > TG, 15.22\%$

4.2. Bloom-level knowledge acquisition

From this point of view we analyze students' learning outcomes for each level evaluated of Bloom's taxonomy, i.e. knowledge, comprehension, application, analysis and synthesis. The ANOVA and Kruskal-Wallis analyses (see table 7) show that significant differences exist among the three groups regarding the analysis and synthesis levels. Tables 8 and 9 show the results of the detailed studies regarding the analysis and synthesis levels respectively. The differences actually exist when comparing either CG or VG with TG. Thus, students from CG and VG obtained better results than those from TG at the analysis (average of 18.55%) and synthesis (average of 17.79%) levels. Therefore, hypotheses H1a and H2a have been confirmed for the analysis and synthesis levels of Bloom's taxonomy; H3a cannot be confirmed.

4.3. Topic knowledge acquisition

From this point of view, we analyze students' learning outcomes for each topic evaluated, i.e. infix operators, user-defined data types and recursive data types. The ANOVA and Kruskal-Wallis analyses (see table 10) show that there are significant differences among the three groups regarding user-defined data types and recursive data types.

Table 9: A detailed comparative study of Bloom-level knowledge acquisition regarding the *Synthesis* level. Cells show the level of significance and the differences (% of improvement) between pairs of groups.

Groups	t-Student analysis	Differences
CG vs. VG	$t(8.784) = -.256, p = .799$	-
CG vs. TG	$t(76) = 2.667$, $p =$ **.009**	$CG > TG, 14.45\%$
VG vs. TG	$t(79.191) = 3.827$, $p <$ **.001**	$VG > TG, 21.13\%$

Table 10: Average grades for each group and topic and ANOVA or Kruskal-Wallis analyses of knowledge acquisition.

	CG	VG	TG	ANOVA or Kruskal-Wallis
Infix operators	.5842	.5296	.4840	$F(2, 123) = 1.153, p = .224$
User-def. data types	.6319	.6289	.3983	$H(2) = 23.260$, $p <$ **.001**
Recursive data types	.7055	.8529	.7923	$F(2, 98) = 8.019$, $p =$ **.001**

Regarding user-defined data types, differences actually exist when comparing either CG or VG with TG, where students from CG and VG improved their results with respect to TG in 23.21% on average (see table 11). Thus, hypotheses H1a and H2a have been confirmed for user-defined data types (a medium complex topic); H3a cannot be confirmed.

The results were different for the topic of recursive data types (a highly complex topic). Now there is a difference of 14.74% (see table 12) between VG and CG. None of the H1a, H2a and H3a hypothesis is confirmed by these results. In fact, the opposite of the H3a hypothesis is confirmed because VG outperformed CG.

Table 11: A detailed comparative study of knowledge acquisition regarding user-defined data types. Cells show the level of significance and differences (% of improvement) between pairs of groups.

Groups	t-Student or Mann-Whitney U analyses	Differences
CG vs. VG	$t(79) = 1.161, p = .249$	-
CG vs. TG	$U = 688.0$, $p <$ **.001**	$CG > TG, 23.36\%$
VG vs. TG	$U = 415.0$, $p <$ **.001**	$VG > TG, 23.06\%$

Table 12: A detailed comparative study of knowledge acquisition regarding recursive data types. Cells show the level of significance and differences (% of improvement) between pairs of groups.

Groups	t-Student analysis	Differences
CG vs. VG	$t(64) = -4.322$, $p <$ **.001**	$VG > CG, 14.74\%$
CG vs. TG	$t(64) = -2.225, p = .030$	-
VG vs. TG	$t(68) = 1.673, p = .099$	-

5. Long-Term Results

In this section we describe the long-term results of the evaluation in terms of knowledge acquisition, drop-out rates and learners' satisfaction. In addition to students' grades we also consider their passing rate. In this section we use a different statistical test suitable for data. Both passing rate and drop-out rate have a binomial nature, therefore we use the binomial test to analyze these data.

5.1. Knowledge acquisition

Knowledge acquisition was measured in terms of the grades of the term exam. None of the students had previous knowledge on the subject. Students' grades have four possible values: failure, fair, good and excellent[1]. The three latter grades allow students to pass the exam.

Regarding these grades, we have detected significant differences among the three groups, $H(6) = 13,408$, $p =$ **.037**. Fig. 3 shows their distribution. Notice that TG is the worst group with 61.1% of failures, GC is the group with more students who passed the term exam with *fair* (35.15%) and *good* (21.65%) grades, and finally VG is the only group with students obtaining *excellent* grades (14%). However, we have not detected significant differences between individual groups.

It seems that the most important difference among groups is passing rate. While CG and VG have similar passing rates, around 55%, TG has a low rate, 38.9%. To test the significance of this difference we use the binomial test of proportions. We know the proportion of students belonging to each group with respect to the whole population of participants; we call

[1]Mapping into the numerical range [0-1]: failure ($grade < 0.5$), fair ($0.5 =< grade < 0.7$), good ($0.7 =< grade < 0.9$) and excellent ($0.9 =< grade =< 1$)

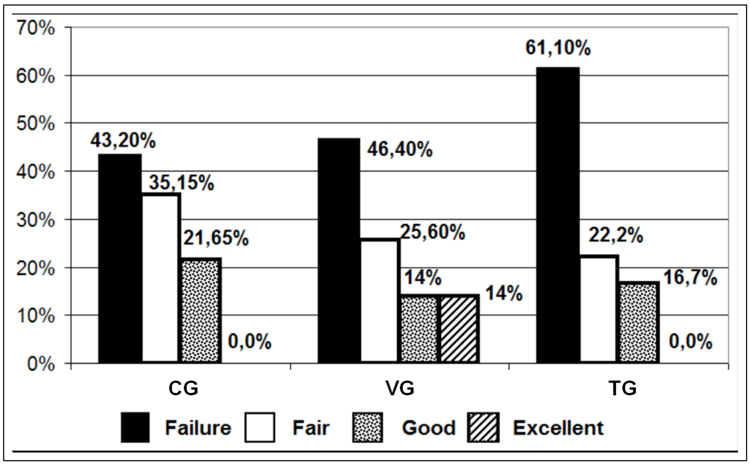

Figure 3: Analysis of the term exam grades among the three groups

it *expected proportion*. With term exam grades, we know the proportion of participants who passed the exam belonging to each group with respect to the whole population of participants who passed the exam; we call it *measured proportion*. If measured and expected proportions of a group are significantly different in terms of the binomial test, then an improvement or worsening has been detected. The joint measured proportion of both groups, CG and VG, is 75.9% while the expected proportion is 61.5%. Both proportions are significantly different, $p = \mathbf{.015}$. Therefore, the joint passing rate of CG and VG, 55.2%, is significantly greater than the passing rate of TG, 38.9%. Therefore, hypotheses H1b and H2b can be confirmed with these data, but H3b cannot be confirmed.

5.2. Students' drop-out rate

Table 13 shows the drop-out rates for all the students enrolled in the course and for those who participated in the evaluation. While the rates are similar for all the students in the three groups, the rates for participants show a possible difference between CG and the other two groups, VG and TG.

Drop-out data have a binomial nature; therefore, we use the binomial test of proportion to analyze these data. The expected proportion for CG is 30.3% while its measured proportion is 16.2%. Expected and measured proportions are significantly different, $p = \mathbf{.041}$. Therefore, CG has significantly decreased its drop-out rate with respect to VG and TG. Thus, hypotheses H1c and H3c

Table 13: Drop-out rate for each group, accounting only the *participants* or *all the students*

	All the students	Participants
CG	34.4% (41/119)	15.0% (6/40)
VG	39.2% (47/120)	32.0% (16/50)
TG	39.6% (59/149)	35.7% (15/42)

are confirmed while H2c cannot be confirmed with these data.

5.3. Learners' satisfaction

We collected students' opinion about animations with a questionnaire (see subsection 3.2.1). Table 14 summarizes students' answers to these questions. Clearly, students' opinion is highly positive. All the participants totally or partially agree with ease of use/construction of animations. In addition, more than 90% in average think that animations are useful and help them understanding the animated concepts.

Table 14: Likert-based questions of the questionnaire. Values represent the % of answers that totally or partially agree with each statement, NA stands for non applicable. The Total column considers CG and VG together.

	CG, N=40	VG, N=50	Total, N=90
Help understanding the concepts	86,2%	100%	93%
Animations are easy to construct	100%	NA	100%
Animations are easy to use	NA	100%	100%
Animations are useful	83,4%	100%	91,5%

6. Discussion

This study evaluates the educational impact of animation construction tasks. Considering animation construction as the treatment under study, we compared it with two control groups: one group with viewing animations tasks, representing a less active approach to algorithm animation, and another group instructed in a traditional way, without the systematic use of animations.

The results of our study confirm some of our hypotheses, so that the educational use of animations improves students' learning outcomes. Due to our detailed analysis we can characterize these learning outcomes in terms of:

the levels of Bloom's taxonomy, topic complexity, passing rate and drop-out rate.

To our knowledge, this is the first study that investigates systematically construction activities and obtains significant successful results. Furthermore, this is the first experiment that consistently measures long-term effects of the use of animation construction activities. Finally, our detailed study of learning outcomes allows proposing a recommended use of the approaches studied in this research. Next we discuss our results following the research questions.

6.1. Question 1: Does the use of program animations improve students' learning outcomes?

The hypotheses derived from this research question, H1 and H2, test whether animation viewing or constructing activities improve learning outcomes with respect to the no-viewing approach.

Hypotheses H1a and H2a (i.e. improvements in short-term knowledge acquisition) were confirmed by our empirical results regarding some of the aspects considered in our study. From a global point of view, no significant differences exist among the three groups. Although we could confirm neither H1a nor H2a from this point of view, it seems that either viewing or constructing animations may improve learning outcomes with respect to a traditional teaching methodology. To check this possibility we performed deeper studies. Then, hypotheses H1a and H2a were confirmed at high levels of Bloom's taxonomy, i.e. the analysis (18.55%) and synthesis (17.79%) levels. In addition, H1a and H2a were confirmed regarding medium complexity topics. The improvement achieved by both uses of animations was 23.21%.

Our results confirmed hypotheses H1b and H2b (i.e. improvements in long-term knowledge acquisition). Students who used animations, either viewing or constructing them, got better grades than students who used the traditional approach. The most representative result is the percentage of students who passed the term exam: 55.2% of students who used animations vs. only 38.9% of students who followed the traditional methodology.

Finally, our results confirm hypothesis H2c (i.e. improvements in dropout rates). Therefore, the number of students who constructed animations and dropped the course out is significantly lower (20.7% less) than those who were taught with the traditional approach.

Therefore, we can confirm that activities for animation viewing or constructing improve students' learning outcomes. They can be used for medium complex topics and their immediate learning effects improve high levels of Bloom's taxonomy –analysis and synthesis. In the long term, they improve passing and drop-out rates.

Finally, and although it is a subjective measure, we may claim that the students' opinion about animations is clearly positive. Most of them (between 83.4% and 100%) believe the following: animations helped them to understand concepts, animations are easy to use and construct, and animations are useful.

6.2. Question 2: Does an increase of students' engagement with program animations lead to improvements in their learning outcomes?

The hypothesis derived from this research question, H3, tests whether animation constructing activities outperform animation viewing activities with respect to learning outcomes.

Our results are mixed. While H3a was rejected for highly complex topics, H3c was confirmed. Therefore, in the short-term, animation viewing tasks seem to be more effective (14.74% more) for dealing with highly complex topics than animation construction tasks. However, in the long-term, animation constructing tasks decrease drop-out rates in 17.0% with respect to animation viewing tasks.

6.3. Hypotheses without Confirmation

Some of the hypotheses of our study were not confirmed. Firstly, learning outcomes were not improved at all the levels of Bloom's taxonomy. It could be that the tasks performed by students with animations were not designed to cope with all levels. We did not initially characterize the tasks with Bloom's taxonomy but they mostly address the analysis level. Consequently, these tasks could not affect the lower levels of the taxonomy.

Secondly, no improvement in learning outcomes was detected from a global point of view. On considering all the levels of Bloom's taxonomy, it could be that improvements detected in analysis and synthesis levels were diminished by the balanced results in the other levels of the taxonomy.

Thirdly, none of the uses of animations improved learning outcomes for simple topics. A plausible explanation could be that simple topics are so easy that the use of animations is not needed.

The rest of the hypotheses without confirmation deals with comparing constructing and viewing activities. We study them in the next subsection.

6.4. The Role of Students' Engagement, Topic Complexity and Textual Contents

The Engagement Taxonomy thesis (Naps et al., 2002) is "the higher the engagement the better the learning outcomes". Hundhausen et al.'s (2002) main conclusion is "what students do with animations has more impact on learning than what visualizations show to the students". Both contributions support active uses of animations. We considered that constructing animations is a more active task than viewing them. Our results partially support this idea. The study shows that animation construction improves learning in terms of drop-out rate. Although drop-out is not an attitude, within our educational context it can be interpreted that if a student takes the term exam, this means that she/he has studied the subject.Unlike other countries, e.g. the USA, students from our country are not required to take the final exam. This is why in the educational context of this study we can associate low drop-out rates with a positive attitude towards a subject. Many of the studies dealing with animation construction detect improvements in student's attitude (Hundhausen and Douglas, 2000; Hundhausen, 2002; Stasko, 1997; Urquiza-Fuentes and Velázquez-Iturbide, 2007). Taking into account these studies and our results, we think that engagement has a significant impact on students motivation.

The impact on knowledge acquisition is not so clear. Neither H3a nor H3b can be confirmed with our results. Furthermore, viewing activities outperformed constructing ones for highly complex topics (H3a) in 14.74%. In addition, only one related study found differences in knowledge acquisition (Hübscher-Younger and Narayanan, 2003), and its results could be influenced by students' motivation prior to the experiment. In our study, all the students had the same motivation to participate in the evaluation.

The question is why a more engaging task did not improve learning outcomes in terms of knowledge acquisition. Moreover, where is the origin of the balanced results? Was the impact of our construction tasks lower than expected or was the impact of our viewing tasks higher than expected? Further research should be carried out on these issues. We analyze here the features of both approaches and their results in order to find possible answers to these questions. Both uses of animations were tested under their best circumstances: animations were constructed with an effortless approach, and the animations viewed were produced by the instructor and enhanced with textual explanations.

The construction approach should improve learning because it uses an effortless approach and it is an active educational task. Using an effortless approach to construct animations allows students to focus on the educational tasks rather than on technical issues (Hundhausen, 2002; Stasko, 1997). In addition, being an active educational task, animation construction should improve the learning outcomes (Hundhausen et al., 2002; Naps et al., 2002).

Let us consider now what *active* means. Literature about *learning* states that meaningful learning requires learner's cognitive activity rather than learner's behavioral activity. This means that just doing things different from listening to the teacher –behavioral activity– is not enough: students must think and reflect about what they are doing –cognitive activity– (Bonwell and Eison, 1991). Note that this means that some behavioral activities could not promote cognitive activity (Mayer, 2005; Prince , 2004). Therefore, active learning tasks should promote learner's cognitive activity (Bonwell and Eison, 1991).

Our construction approach is effortless from a behavioral point of view. In terms of the Cognitive Load Theory (Sweller et al., 1998), effortlessness reduces *extraneous* cognitive load. The cognitive effort needed to produce an animation depends on the concepts to animate –*intrinsic* cognitive load (Sweller et al., 1998). Students' cognitive work consists of: (1) achieving a deep understanding of how the program solves the problem, (2) choosing suitable input data that produce a meaningful animation, (3) identifying the animation steps that better describe the program behavior to include them in the animation, and (4) producing a meaningful explanation of how the program solves the problem. These four steps are focused on *germane* cognitive load, which is devoted to the processing, construction and automation of schemas (Sweller et al., 1998).

In our study, the students who constructed animations for highly complex topics obtained worse grades than those who viewed animations. In our opinion, the cognitive effort required to produce animations for these concepts was too high because there was too much *intrinsic* cognitive load. It exceeded the students' possibilities because it decreased students' attention to *germane* cognitive load, thus causing low grades in the knowledge test. Therefore, the educational impact of our construction tasks was lower than expected for highly complex topics.

However, the balanced results between viewing and constructing tasks for medium complex topics are not explained by the previous rationale. Now let us focus on the viewing tasks. Our viewing approach could improve learning because the animations were developed by the instructors and were accompanied by educational features (see section 2) with an important impact on educational effectiveness. In addition, the animations were enhanced with textual explanations, also produced by the instructor.

The educational impact of textual explanations has not been completely studied in the literature (Hundhausen et al., 2002), and we think that it may play an important role to explain our results. Firstly, their use is based on the *Dual Coding Theory* (Paivio, 1983). Mayer (2005) states that integrating narrative or textual contents in multimedia materials should improve learning if they are used according to some principles. Secondly, there are two related studies (Kumar, 2005; Lawrence, 1993) that report learning improvements in terms of knowledge acquisition when animations are enhanced with some kind of textual contents. Moreover, the only study that does not mention the use of textual contents (Stasko, 1997) failed to find significant results regarding knowledge acquisition. Finally, explanations have produced positive results in other scopes like collaborative learning (Laakso et al., 2009).

Our animation viewing tasks asked students to ensure that they understood each animation step. In our opinion, the textual explanations produced by the teacher and integrated into the animations supported this task and promoted the students' cognitive activity which is needed to learn the concepts animated. Although our viewing approach is a less active task from a behavioral point of view, it is an active task from a cognitive point of view. Therefore, our viewing approach achieved similar results than our constructing approach when used for medium complex topics, and better when used for highly complex topics. Consequently, the educational impact of our viewing tasks was higher than expected.

6.5. Recommended use of animations

Based on our empirical results we can derive a methodological approach to the educational use of animations. The three approaches considered in this study can be effectively used. Firstly, sets of animations could be facilitated for self-study. Secondly, theoretical sessions would be given as lectures with examples and especially prepared animations. Finally, laboratory sessions would mix the traditional, viewing and constructing approaches depending on the complexity of the concepts involved.

For simple concepts, the traditional approach is enough. These sessions can be used to introduce the programming language and the development environment as well. When working with medium complex concepts, construction tasks similar to those used in this study would be used. Finally, for highly complex concepts the core of the sessions would be formed by the traditional problem-solving approach supported by the teacher and ready-to-play animations. In this case, animation construction could also be used but as a secondary task, e.g. asking students to document their solutions with explanatory animations.

7. Conclusion

Previous research on algorithm and program visualization suggests that active approaches are educationally effective (Hundhausen et al., 2002; Naps et al., 2002). This result also is supported by learning theories such as Cognitive Constructivism or Active Learning (Bonwell and Eison, 1991). In this work we studied the educational impact of an active and highly engaging approach, namely the construction of program animations by students. This is one of the first analyzes of animation construction activities using a systematic approach, where it is compared with other educational uses and significant learning improvements are measured. Moreover, it is the first study that investigates the long-term effects of animation construction activities.

Educational uses of animations should improve students' learning outcomes. Therefore, active educational uses of animation (e.g. animation construction) should improve learning outcomes with respect to other educational uses (e.g. animation viewing). We have conducted an experiment to gather supporting evidence on these claims. We compared the program animation construction approach (the treatment group) with two possible scenarios (two control groups): a less active approach, where students had to view animations, and a traditional approach where animations are not used. Note that the viewed animations had been carefully designed and developed by the instructors.

Short-term results indicate a possible advantage of both uses of animation over the traditional approach. This advantage is based on improvements at the analysis and synthesis levels of Bloom's taxonomy. However, results vary depending on the complexity of the topics covered. No difference was detected for simple topics. For medium complex topics, both uses of animations delivered better results than the traditional approach. Finally, for highly complex topics, viewing animations outperformed the construction approach.

Long-term results also show improvements when animations are used. Knowledge acquisition was similarly improved by both uses of animations regarding term exam passing-rate. In addition, construction tasks significantly improved student's motivation with respect to both the viewing and the traditional approaches. Finally, students' opinion is clearly positive towards the use of animations.

Based on these results we conclude that increasing students' engagement with animations can significantly decrease drop-out rate. This can be explained by Cognitive Constructivism or Active Learning (Bonwell and Eison, 1991) theories. But there is not empirical evidence to support that animation construction tasks improve students' knowledge acquisition more than animation viewing tasks. Textual explanations integrated into the program animations and topic complexity seem to have an important impact on knowledge acquisition. They can be supported by Dual Coding Theory (Paivio, 1983) and Cognitive Load Theory (Sweller et al., 1998). In fact, viewing animations enhanced with textual explanations is the best of the approaches we evaluated to teach highly complex topics.

Appendix A. Knowledge tests

In this appendix we provide an example of the knowledge test that students completed during the short-terms evaluations. The questions of these tests are mapped into some levels of Bloom's taxonomy (Bloom et al., 1959). In this knowledge test, each question is also identified with its corresponding level.

1. (Knowledge level) What is the `type` sentence for?
2. (Knowledge level) What is the `data` sentence for?
3. Given the following Hope program:

```
data INTERVAL ==
  Edg (real # real) ++  ! Edges
  Cen (real # real);    ! Center and radius

type INTERVAL_INTEGER == num # num;
dec Inside : INTERVAL # real -> truval;
--- Inside(Edg(i,s) , p) <= (p >= i) and (p =< s);
--- Inside(Cen(c,r) , p) <= (p >= (c-r)) and (p =< (c+r));
```

```
dec Contains : INTERVAL # INTERVAL -> truval;
--- Contains(Edg(i1,s1) , Edg(i2,s2)) <=
    (Inside(Edg(i1,s1),i2) and Inside(Edg(i1,s1),s2));
```

(a) (Understanding level) Copy the part of the source code where the user-defined types only **rename** other data types.
(b) (Understanding level) Copy the part of the source code where the user-defined types **create** new data types.
(c) (Application level) Write the result of executing the following expression:
`Contains(Edg(4.0*3.2 , 25.1) , Edg(13.0,20.9));`
(d) (Application level) Write a Hope expression whose execution satisfies the following requirements:
   - The function `Contains` must be executed during the evaluation of the expression.
   - Some of the Boolean calculations performed during the evaluation of the function `Contains` must return `true`.
   - The result of the whole evaluation of the expression must be **false**.

4. Complex numbers have two components, the real part and the imaginary part. The real part is represented by a real number, while the imaginary part is represented by the multiplication of a real number by the imaginary unit i (where $i = \sqrt{-1}$). By convention, complex numbers are written as $a + bi$, where a is the real part and bi the imaginary part.
   Complex numbers can be operated in many ways, but here we only specify three operations:

   Addition: $(a + bi) + (c + di) = (a + c) + (b + d)i$

   Multiplication: $(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$

   Modulus: Let be $m = a + bi$; $|m| = \sqrt{a^2 + b^2}$

   (a) (Analysis level) Copy the part of the problem statement that gives information about a new data type that should be created with the `data` sentence.
   (b) (Analysis level) Copy the part of the problem statement that gives information about the use of the data type as parameters in a function call.

(c) (Analysis level) Copy the part of the problem statement that gives information about the use of the data type as the value returned by a function.

(d) (Synthesis level) Develop a Hope program that implements the data type and the operations described.

Appendix B. Term Exam

1. **Lab exercise.** The following Hope program computes how to give the change of a purchase into a minimum number of notes of 10 and 5 euros, and coins of 2 and 1 euro:

```
dec change : num # num -> num # num # num # num;
--- change (paid,price) <=
  if paid =< price
  then (0,0,0,0)
  else ((paid-price) div 10 , (paid-price) mod 10 div 5 ,
        (paid-price) mod 10 mod 5 div 2 ,
        (paid-price) mod 10 mod 5 mod 2);
```

   Optimize the previous program as much as possible using local definitions.

2. **Theoretical question.** In the context of functional programming with Hope, what is polymorphism? Explain your answer.

3. **Problem.** Using the Hope programming language, write a data type to manage sets of integer numbers. The name of the data type must be "SET". You are not allowed to use the built-in type list of Hope. The set type allows storing an unlimited number of integers (all of them distinct), even none.
   In addition, it is required to write a Hope program with the following functions and operators:

   Cardinal: a function that returns the number of elements in a set (passed as a parameter).

   Belongs: a function that returns `true` if a given number (passed as a parameter) exists in the set (passed as a parameter).

   Infix operators: `union`, `intersection` and `subtraction`:

- The `union` operator will have lower priority than the `intersection` operator. In addition, `subtraction` will have the same priority as `union`.
- Note that the subtraction "a minus b" does not satisfy the commutative property.

References

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education 33* (2-3), 131–152.

Anderson, J., & Naps, T. (2001). A context for the assessment of algorithm visualization system as pedagogical tools. In E. Sutinen (Ed.), *Proceedings of the First Program Visualization Workshop* (pp. 121–130). Joensuu, Finland: University of Joensuu.

Baecker, R. (1998). Sorting Out Sorting: A case study of software visualization for teaching computer science. In J. Stasko, J. Domingue, M. Brown, & B. Price (Eds.), *Software Visualization. Programming as a Multimedia Experience* (pp. 369–381). Cambridge, MA: MIT Press.

Ben-Ari, M. (2001). Program visualization in theory and practice. *Informatik/Informatique 2*, 8–11.

Ben-Bassat, R., Ben-Ari, M., & Uronen, P.A. (2003). The Jeliot 2000 program animation system. *Computers & Education 40* (1), 1–15.

Bloom, B., Furst, E., Hill, W., & Krathwohl, D. (1959). *Taxonomy of Educational Objectives: Handbook I, The Cognitive Domain.* New York, NY: Longman.

Bonwell, C.C. & Eison, J.A. (1991). Active learning: Creating excitement in the classroom. *ASHEERIC Higher Education Report No. 1*, George Washington University, Washington, DC.

Crosby, M., & Stelovsky, J. (1995). From multimedia instruction to multimedia evaluation. *Journal of Educational Multimedia and Hypermedia 4*, 147–162.

De Landsheere, V. (1997). Taxonomies of educational objectives. In J.P. Keeves (Ed.), *Educational Research, Methodology and Measurement: An International Handbook* (pp. 803–812). Oxford, UK: Pergamon.

Flavell, J. (1989).Introduction. In L.B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser* (pp. 1–24). Hillsdale, NJ: Erlbaum.

Grissom, S., McNally, M.F., & Naps, T.L. (2003). Algorithm visualization in CS education: Comparing levels of student engagement. In S. Diehl & J.T. Stasko (Eds.), *Proceedings of the 2003 ACM Symposium on Software Visualization* (pp. 87–94). New York, NY: ACM Press.

Hübscher-Younger, T., & Narayanan, N. (2003). Dancing hamsters and marble statues: characterizing student visualizations of algorithms. In S. Diehl & J.T. Stasko (Eds.), *Proceedings of the 2003 ACM Symposium on Software Visualization* (pp. 95–104). New York, NY: ACM Press.

Hundhausen, C. (2002). Integrating algorithm visualization technology into an undergraduate algorithms course: Ethnographic studies of a social constructivist approach. *Computers & Education 39* (3), 237–260.

Hundhausen, C., & Brown, J. (2008). Designing, visualizing, and discussing algorithms within a CS 1 studio experience: An empirical study. *Computers & Education 50* (1), 301–326.

Hundhausen, C.D. & Douglas, S.A. (2000). Using visualizations to learn algorithms: Should students construct their own, or view an expert's? In *Proceedings. 2000 IEEE International Symposium on Visual Languages* (pp. 21–28). Los Alamitos, CA: IEEE Computer Society Press.

Hundhausen, C., Douglas, S., & Stasko, J. (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing 13* (3), 259–290.

Kann, C., Lindeman, R., & Heller, R. (1997). Integrating algorithm animation into a learning environment. *Computers & Education 28* (4), 223–228.

Kehoe, C., Stasko, J., & Taylor, A. (2001). Rethinking the evaluation of algorithm animations as learning aids: An observational study. *International Journal of Human-Computer Studies 54* (2), 265–284.

Kumar, A. (2005). Results from the evaluation of the effectiveness of an online tutor on expression evaluation. In W. Dann, T.L. Naps, P. Tymann & D. Baldwin (Eds.), *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (pp. 216–220). New York, NY: ACM Press.

Laakso, M.-J., Myller, N., & Korhonen, A. (2009). Comparing learning performance of students using algorithm visualizations collaboratively on different engagement levels. *Educational Technology & Society 12* (2), 267–282.

Larkin, J., & Simon, H.A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science 11* (1), 65–99.

Lauer, T. (2008). Reevaluating and refining the engagement taxonomy. In J. Amillo & C. Laxer (Eds.), *Proceedings of the 13th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 355–355). New York, NY: ACM Press.

Lawrence, A. (1993). Empirical studies of the value of algorithm animation in algorithm understanding. Ph.D. thesis, Dep. of Computer Science, Georgia Institute of Technology.

Mayer, R. (2005). *Multimedia Learning*. Cambridge University Press.

Myller, N., Bednarik, R., Sutinen, E. & Ben-Ari, M. (2009). Extending the engagement taxonomy: Software visualization and collaborative learning. *ACM Transactions on Computing Education 9*, 7:1–7:27.

Naps, T., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., et al. (2002). Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin , Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education 35* (2), 131–152.

Paivio, A. (1983). *The Empirical Case for Dual Coding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Pareja-Flores, C., Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2007). WinHIPE: an IDE for functional programming based on rewriting and visualization. *ACM SIGPLAN Notices 42* (3), 14–23.

Patterson, R. (2010). Hope. http://www.soi.city.ac.uk/~ross/Hope/.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education 93* (3), 223–232.

Saraiya, P., Shaffer, C. A., McCrickard, D. S., & North, C. (2004). Effective features of algorithm visualizations. In D. Joyce & D. Knox (Eds.), *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (pp. 382–386). New York, NY: ACM Press.

Scaife, M., & Rogers, Y. (1996). External cognition: How do graphical representations work? *International Journal of Human-Computer Studies 45* (2), 131–152.

Stasko, J. (1997). Using student-built algorithm animations as learning aids. In J.E. Miller (Ed.), *Proceedings of the Twenty-Eighth SIGCSE Technical Symposium on Computer Science Education* (pp. 25–29). New York, NY: ACM Press.

Stasko, J., Domingue, J., Brown, M., & Price, B. (1998). *Software Visualization. Programming as a Multimedia Experience*. Cambridge, MA: MIT Press.

Sweller, J., Van Merriënboer, J. & Paas, F. (1998). Cognitive architecture and instructional design. *Educational Psychology Review 10* (3), 251-296.

Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2005). Effortless Construction and Management of Program Animations on the Web. R.W.H. Lau, Q. Li, R. Cheung & W.Liu (Eds.), *Advances in Web-Based Learning ICWL 2005, 4th International Conference, Proceedings* (pp. 163-173). Berlin, Heidelberg: Springer-Verlag.

Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2007). An evaluation of the effortless approach to build algorithm animations with WinHIPE. *Electronic Notes in Theoretical Computer Science 178*, 3–13.

Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2009). A survey of successful evaluations of program visualization and algorithm animation systems. *ACM Transactions on Computing Education 9*, 9:1–9:21.

Velázquez-Iturbide, J. Á., Pareja-Flores, C., & Urquiza-Fuentes, J. (2008). An approach to effortless construction of program animations. *Computers & Education 50* (1), 179–192.

Velázquez-Iturbide, J. Á., Sánchez-Calle, A., Duarte-Muñoz, A., & Lázaro-Carrascosa, C. (2005). *Ejercicios Resueltos de Bases de Lenguajes de Programación*. Madrid: Editorial Universitaria Ramón Areces.