

Empirical Evaluation of Usability of Animations in a Functional Programming Environment

Technical Report 141/04

Departamento de Sistemas Informáticos y Programación

Universidad Complutense de Madrid

Spain

M^a Ángeles Medina Sánchez¹, Carlos Alfredo Lázaro Carrascosa³,
Cristóbal Pareja Flores², Jaime Urquiza Fuentes³ and J. Ángel Velázquez Iturbide³

¹ Departamento de Estadística e Investigación Operativa III, Escuela Universitaria de Estadística, Universidad Complutense de Madrid, Spain

² Departamento de Sistemas Informáticos y Programación, Escuela Universitaria de Estadística, Universidad Complutense de Madrid, Spain. cpareja@sip.ucm.es

³ Departamento de Informática, Estadística y Telemática, Escuela Superior de Ciencias Experimentales y Tecnología, Universidad Rey Juan Carlos, Spain. {carlos.lazaro,jaime.urquiza,angel.velazquez}@urjc.es

1. Introduction

Algorithm animations have been used for teaching computer programming since the birth of the field in the late seventies. One of the most notable obstacles to their adoption is the considerable effort that the production of program animations represents for the instructor. To foster the use of animations among students, we have extended the integrated programming environment (IDE) WinHIPE with visualization and animation capabilities [1]. The environment has been used for teaching functional programming at the Universidad Rey Juan Carlos in the last two years.

In the academic year 2001/2002, an experiment was made [2] to compare two versions of the same IDE: TurboHIPE (a textual IDE) and WinHIPE 1.0 (a graphical IDE). Results showed that students who used WinHIPE were enthusiastic with animations but they did not learn more than those who used TurboHIPE.

We have conducted an experiment to obtain empirical evidence about whether WinHIPE 2.0 allows building animations easily and with minimal workload. We have also analyzed whether animations are easy to use and whether they are perceived by the students as an aid for other tasks. We describe in this report the experiment in detail, as well as its results.

2. The Animation Construction Process

To build an animation, the user just performs the same procedure as to execute a program: first, a (functional) program is compiled, and then it is executed (by evaluating an expression). The evaluation is typically performed by means of a sequence of actions. Each action produces a progress in the evaluation status of the expression, yielding an intermediate expression. The evaluation ends when the final value is obtained. Each intermediate expression can be represented either graphically or textually (an example of both formats can be seen in figure 1). The external representation of an expression provides the user with a static visualization (or visualization, for short).

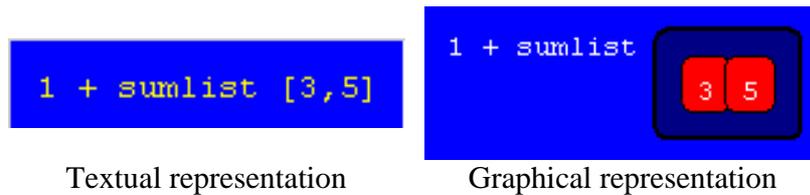


Figure 1. Textual and graphical representation of expressions

Once the user has produced all the visualizations (i.e. the evaluation of the expression is finished), he/she can play the animation using a simple VCR interface (see figure 2).

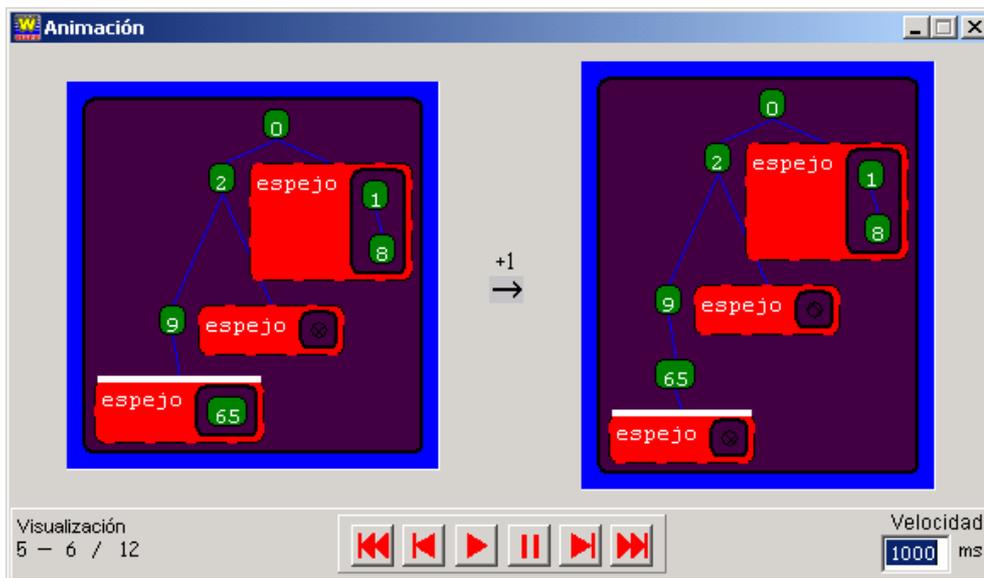


Figure 2. Interface to play animations

An animation is built by selecting the visualizations that will be shown. Visualizations are accumulated in the evaluation window, with a checkbox placed at the left hand-side of each visualization. Thus, the user selects each visualization by checking its corresponding checkbox. Frequently, the number of visualization exceeds the space available in the evaluation window. Another window (namely, the miniatures window) is intended to make simultaneously visible as many visualizations as possible, each one in reduced size. An

example of both windows is shown in figure 3 (the evaluation window at the left and the miniature window at the right).

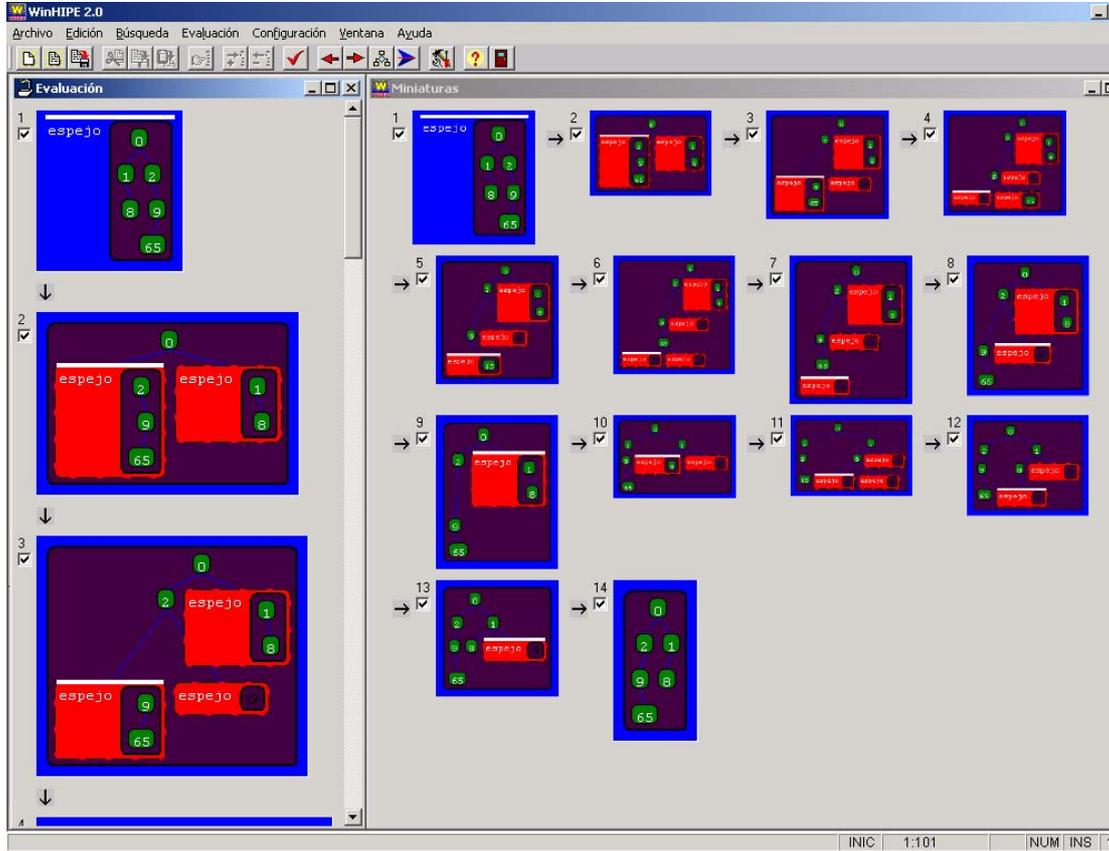


Figure 3. The evaluation and the miniature windows

3. Experiment

We have conducted a controlled experiment to make an empirical evaluation of usability of animations in WinHIPE. Subjects had to perform some tasks where animations are potentially useful, but their use is optional.

3.1. Hypotheses

We tested six hypotheses about WinHIPE, three of them having guided the design of the experiment. They follow:

1. Animations are easy to use and useful (hypothesis 1).
2. The animation construction process is easy to learn. We analyze whether students use animations (hypothesis 5) and whether they think that animations are easy to use (hypothesis 2). Both hypotheses are related to the previous experience of the student using the animation capabilities of WinHIPE.

3. Not only animations are perceived as a goal, but they are also perceived as an aid for other tasks. We analyze whether previous experience with WinHIPE facilitates the understanding (hypothesis 3) and the debugging (hypothesis 4) of algorithms. We also analyze whether students who think that animations are easy to use, actually use them (hypothesis 6).

3.2. Subjects

Fifty-two subjects took part in the experiment. The subjects were first course students of computer science at the Universidad Rey Juan Carlos. Participation in the experiment was voluntary. The course schedule includes six laboratory sessions during the term plus one for an assignment exam. We performed the experiment during the fifth laboratory session.

Hypotheses two and five did not require users to have previous experience with animation capabilities of WinHIPE. In fact, 46.2% of the subjects had never used such capabilities, 30.8% had sporadically used them, 15.4% frequently and 7.6% had always used them with WinHIPE.

3.3. Tasks

The tasks performed during the experiment session were designed to meet the most important use cases of animation: program comprehension, development and debugging. The full assignment can be found in Appendix A.

The program comprehension task consists in building an animation of an algorithm. Three steps were followed. Firstly, the teacher explained a problem. Secondly, a solution to the problem was given to the students. Finally, the students were asked to build an animation of the solution, without any further explanation.

The program development task consists in constructing a solution to a problem similar to the previous one.

The program debugging task consists in locating, fixing and explaining five errors in a buggy program. The students were not forced to use the animation capabilities of WinHIPE.

3.4. Procedure

The experiment was conducted in the same laboratory where the students performed their lab sessions. They used Intel Pentium III personal computers, with 933MHz, 128Mb and a 17" screen. These PCs are powerful enough to execute the programming environment and to deal with the visualization and animation capabilities of the environment.

Upon arriving at the lab, subjects were informed about the experiment (why do we do it?, what do we want to know?). The experiment was divided into three parts: an animation part, a debugging part and a questionnaire:

1. The animation part begins with a fifteen minutes explanation of the animation construction process. Then, a problem is described and its solution is given. The

- subjects are required to build an animation in twenty minutes. Finally, a problem analogous to the previous one is posed, and they had twenty-five minutes to solve it.
2. In the debugging part, the teacher first debugs a program using WinHIPE without using animations; this takes five minutes. Then a program with five errors is given and the students have to find them. For each error they have: to locate the error (identifying the line number and the buggy expression), to write a correct version of the program, and to explain their solution. They have twenty minutes to deal with all the errors.
 3. Finally, students have to answer a questionnaire.

3.5. Experiment Design and Variables Used

As we have described, the subjects were told to answer a questionnaire after performing the tasks of the experiment. It can be found in Appendix B. We extracted from this questionnaire the values of the eight following variables:

- Previous use of WinHIPE.
- Use easiness of animations.
- Comprehension of the algorithm to which they have to build an animation.
- Utility of animation in the comprehension of the previous algorithm.
- Animations helps in solving the problem proposed.
- Comprehension of the algorithm that they have to debug.
- Utility of animations in learning algorithms.
- Use easiness of WinHIPE.

4. Results and Conclusions

We have used Statgraphics, v. 4.0 to analyze the answers to the questionnaire. Detailed information about the statistic process can be found in Appendix B. We summarize here the results for each hypothesis.

From hypothesis one we can conclude that animations in WinHIPE are easy to use and useful; in fact, more than 60% of students answered that animations were easy to use, whilst less than 8% think that animations are difficult to use. We also feel that students agree that animations assist them in understanding programs. In fact, more than 42% of the students agreed that animations were very useful to understand programs, while less than 8% consider that animations are not useful to understand programs.

From hypotheses two and five we can conclude that the animation construction process is easy to learn and useful for debugging because students have used animations and think that animations are easy to use, independently of their previous experience with the environment.

From hypotheses three and four we can conclude that students understand animations not only as another exercise, i.e. “build an animation for this algorithm...”, but they understand animations as an aid to perform other tasks because animations used them to debug. Hypothesis 6 couldn't be statistically checked with the current experiment.

Appendix A: The Assignment

The assignment consisted in two exercises.

Exercise 1.

Given two integers, the following function must deliver the greatest one. In case they are equal, the function must deliver one number.

The function is a low-level implementation, where the predicates '>' and '>=' can only be used with numbers of one digit.

The function contains 5 different errors. The student must locate, fix and explain them, according to the procedure explained.

```
dec greatestAux: num # num # num -> num;
--- greatestAux (m,n,a) <=
    if (m div 10 > 0) and (n>=10)
    then 2
    else
        if (n<0) and (m>9)
        then 1
        else
            if (m<10) and (n<10)
            then
                if (m>n)
                then 1
                else
                    if (n<m)
                    then 2
                    else a
            else
                if (m mod 10) > (n mod 10)
                then greatestAux (m div 10, n, 1)
                else
                    if (m mod 10) < (n mod 10)
                    then greatestAux (m div 10, n, a)
                    else greatestAux (m div 10, n ,2);

dec greatest: num # num -> num;
--- greatest (m,n) <=
    let b == greatestAux(m,n,0) in
    if b = 1
    then m
    else
        if b = 2
        then n
        çelse 0;
```

Exercise 2. Part (a)

The student must analyze the algorithm and try to understand it. In addition, he/she must generate one or several animations that accurately show the essentials of the algorithm and assist him/her in understanding it.

```
dec Multiplo3Aux: num#num -> truval;
--- Multiplo3Aux (n,a) <=
    if n=0
    then
        if (a<10) and (a>=0)
        then (a=3) or (a=6) or (a=9)
        else Multiplo3Aux (a,0)
    else Multiplo3Aux (n div 10, a + n mod 10);

dec Multiplo3 : num -> truval;
--- Multiplo3 n <= Multiplo3Aux (n,0);
```

Exercise 2. Part (b)

Code a function that decides whether a number is a divisor of 3. The technique of accumulating parameters must be used. It is not allowed to use the built-in functions *div* and *mod*, except with numbers of one digit.

Obviously, the algorithm must be different from the algorithm before.

HINT: The accumulator can be used to store remainders.

Appendix B: The Questionnaire

(The names of the questions are included to relate them to their subsequent statistical analysis.)

- **[Q1 – “uno” (one)]** Have you ever used the visualization and animation capabilities of WinHIPE? (mark with an X).

Never	
Sporadically	
Frequently	
Always	

Please, answer the following questions with a **number between 0 and 4**, taking into account the meaning of each number (underline the selected answer):

- **[Q2 – “dos” (two)]** Were animations easy to use?
0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)
- **[Q3 – “tres” (three)]** Did you understand the algorithm of the exercise when you were told to build an animation?
0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)
- **[Q4 – “cuatro” (four)]** Was the animation useful to understand the algorithm?
0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)
- **[Q5 – “cinco” (five)]** Did the building of the animation assist you in solving the subsequent problem?
0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)

- [Q6 – “seis” (six)] Did you understand the algorithm of the exercise when you were told to debug a program?

0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)

- [Q7 – “siete” (seven)] Did you use animations to debug the algorithm?

0 (no) 1(yes)

- [Q8 – “ocho” (eight)] Independently from the fact that animations can be improved, do you think that they are useful to learn algorithms?

0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)

- [Q9 – “nueve” (nine)] In general, do you think that the programming environment was easy to use?

0(totally disagree) 1(disagree) 2(without opinion) 3(agree) 4(totally agree)

Please, answer briefly to these questions:

- What aspects of animations do you consider positive?

Write down **the most important** for you.

- What aspects of animations do you consider negative?

Write down **the most important** for you.

Appendix C: Statistical analysis

In this section, we include the statistical procedure applied to each hypothesis and our main conclusions. Notice that some of the hypotheses couldn't be confirmed with the present experiment, as we told previously.

(The software used to obtain this study was Statgraphics, v. 4.0.)

Hypothesis 1: "Animations are easy to use, and useful."

Statistical analysis performed: Discriminant analysis, and uni-sample analysis.

(a) Discriminant analysis

This procedure is designed to develop a set of discriminating functions which can help predict uno based on the values of other quantitative variables. 52 cases were used to develop a model to discriminate among the 4 levels of uno. 8 predictor variables were entered. None of the discriminating functions have P-values less than 0.05, so that none are statistically significant at the 95% confidence level. To plot the discriminating functions, select Discriminant Functions from the list of Graphical Options. To predict new observations, select Classification Table from the list of Tabular Options.

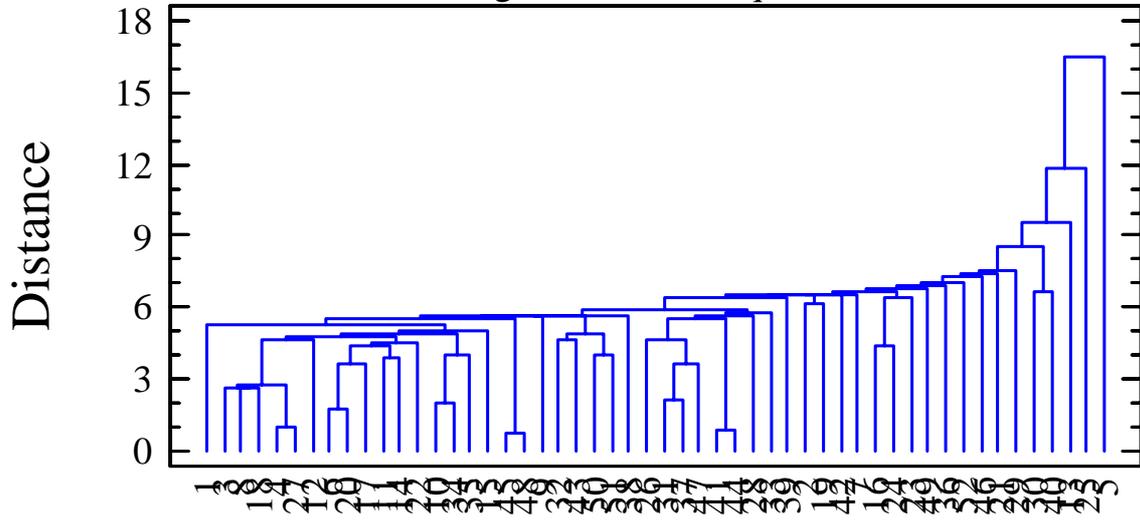
Discriminant Function	Eigenvalue	Relative Percentage	Canonical Correlation
1	0,3233	49,77	0,49428
2	0,265905	40,94	0,45831
3	0,0603665	9,29	0,23860

Functions Derived	Wilks Lambda	Chi-Square	DF	P-Value
1	0,562969	25,8538	24	0,3606
2	0,744977	13,2481	14	0,5071
3	0,94307	2,6377	6	0,8528

This procedure has created 1 cluster from the 52 observations supplied. The clusters are groups of observations with similar characteristics. To form the clusters, the procedure began with each observation in a separate group. It then combined the two observations which were closest together to form a new group. After recomputing the distance between the groups, the two groups then closest together were combined. This process was repeated until only 1 group remained. To specify the number of final clusters, press the alternate mouse button and select Analysis Options. To determine a reasonable value for the number of clusters, look at the Agglomeration Distance Plot available from the list of Graphical Options.

Dendrogram

Nearest Neighbor Method, Squared Euclidean

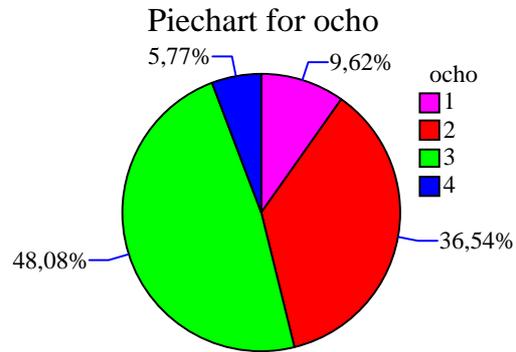


(b) Uni-sample analysis

Frequency Table for ocho

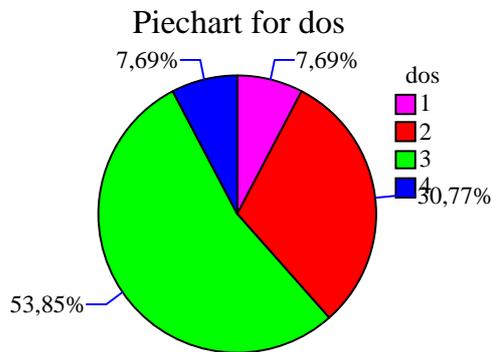
Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	1	5	0,0962	5	0,0962
2	2	19	0,3654	24	0,4615
3	3	25	0,4808	49	0,9423
4	4	3	0,0577	52	1,0000

This table shows the number of times each value of ocho occurred, as well as percentages and cumulative statistics. For example, in 5 rows of the data file ocho equaled 1. This represents 9,61538% of the 52 values in the file. The rightmost two columns give cumulative counts and percentages from the top of the table down.



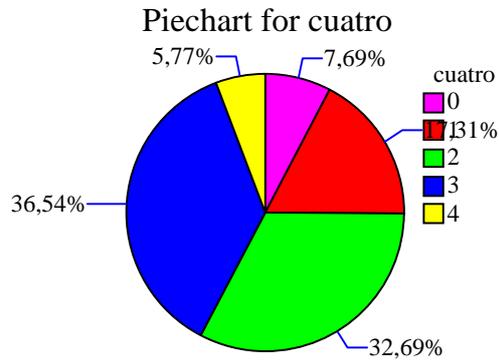
Frequency Table for dos

Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	1	4	0,0769	4	0,0769
2	2	16	0,3077	20	0,3846
3	3	28	0,5385	48	0,9231
4	4	4	0,0769	52	1,0000



Frequency Table for cuatro

Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	0	4	0,0769	4	0,0769
2	1	9	0,1731	13	0,2500
3	2	17	0,3269	30	0,5769
4	3	19	0,3654	49	0,9423
5	4	3	0,0577	52	1,0000



Our conclusion: We notice that students are merged, independently of their previous experience using WinHipe, and also independently of their comprehension of the assigned algorithms. Furthermore, we observe that means for variables “ocho”, “dos” and “cuatro” are high.

Hypothesis 2: “Students find animations easy to use, independently of whether they have used previously the environment or not.”

Statistical analysis used: Multiple-Sample Comparison.

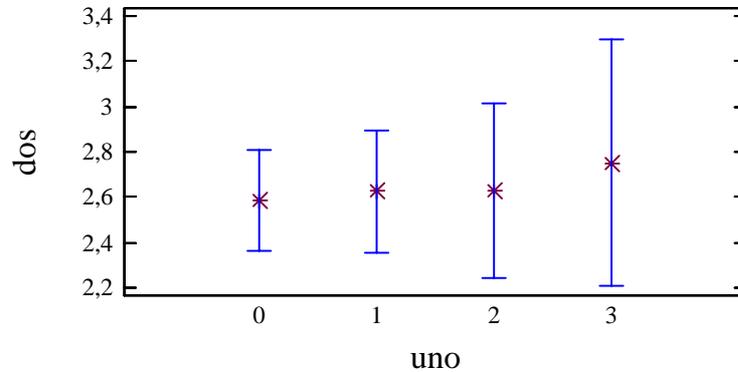
Summary Statistics for dos

uno	Count	Average
0	24	2,58333
1	16	2,625
2	8	2,625
3	4	2,75
Total	52	2,61538

uno	Minimum	Maximum
0	1,0	4,0
1	1,0	4,0
2	2,0	3,0
3	2,0	3,0
Total	1,0	4,0

This table shows various statistics for each of the 4 columns of data. To test for significant differences amongst the column means, select Analysis of Variance from the list of Tabular Options. Select Means Plot from the list of Graphical Options to display the means graphically.

Means and 95,0 Percent LSD Intervals



ANOVA Table for dos by uno

Analysis of Variance					
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Between groups	0,099359	3	0,0331197	0,06	0,9822
Within groups	28,2083	48	0,587674		
Total (Corr.)	28,3077	51			

The ANOVA table decomposes the variance of the data into two components: a between-group component and a within-group component. The F-ratio, which in this case equals 0,0563572, is a ratio of the between-group estimate to the within-group estimate. Since the P-value of the F-test is greater than or equal to 0,05, there is not a statistically significant difference between the means of the 4 variables at the 95,0% confidence level.

Box-and-Whisker Plot

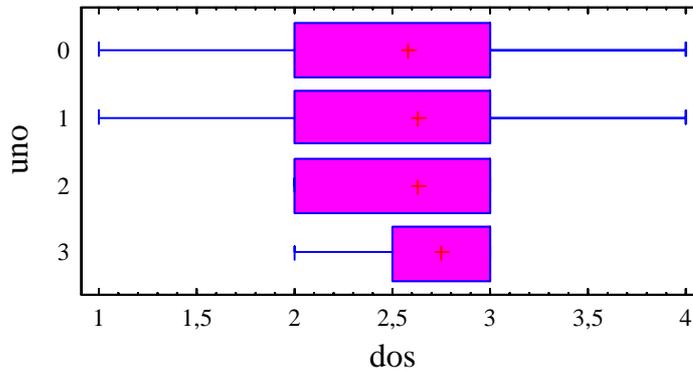


Table of Means for dos by uno with 95,0 percent LSD intervals

uno	Count	Mean	Std. error (pooled s)	Lower limit	Upper limit
0	24	2,58333	0,156481	2,36086	2,80581
1	16	2,625	0,19165	2,35252	2,89748
2	8	2,625	0,271034	2,23966	3,01034
3	4	2,75	0,383299	2,20505	3,29495
Total	52	2,61538			

This table shows the mean for each sample. It also shows the standard error of each mean, which is a measure of its sampling variability. The standard error is formed by dividing the pooled standard deviation by the square root of the number of observations at each level. The table also displays an interval around each mean. The intervals currently displayed are based on Fisher's least significant difference (LSD) procedure. They are constructed in such a way that if two means are the same, their intervals will overlap 95,0% of the time. You can display the intervals graphically by selecting Means Plot from the list of Graphical Options. In the Multiple Range Tests, these intervals are used to determine which means are significantly different from which others.

Kruskal-Wallis Test for dos by uno

uno	Sample Size	Average Rank
0	24	26,0833
1	16	26,625
2	8	26,25
3	4	29,0

Test statistic = 0,159999 P-Value = 0,983773

The Kruskal-Wallis test tests the null hypothesis that the medians within each of the 4 columns is the same. The data from all the columns is first combined and ranked from smallest to largest. The average rank is then computed for the data in each column. Since the P-value is greater than or equal to 0,05, there is not a statistically significant difference amongst the medians at the 95,0% confidence level.

Our conclusion: Animations are easy to use.

A possible interpretation of this conclusion is that animations are so easy to use with WinHipe that it doesn't matter whether students had been previously used: they always consider animations easy to use.

Hypothesis 3: "The previous use of WinHIPE helps to understand the algorithm to be animated."

Statistical analysis used: Multiple-Sample Comparison.

Multiple-Sample Comparison

Summary Statistics for tres

uno	Count	Average
0	24	2,33333
1	16	2,8125
2	8	2,875
3	4	3,0
Total	52	2,61538
uno	Minimum	Maximum
0	0,0	4,0
1	1,0	4,0
2	2,0	4,0
3	3,0	3,0
Total	0,0	4,0

WARNING: There is more than a 3 to 1 difference between the smallest standard deviation and the largest. This may cause problems since the analysis of variance assumes that the standard deviations at all levels are equal. Select Variance Check from the list of Tabular Options to run a formal statistical test for differences among the sigmas. You may want to consider transforming the data to remove any dependence of the standard deviation on the

mean.

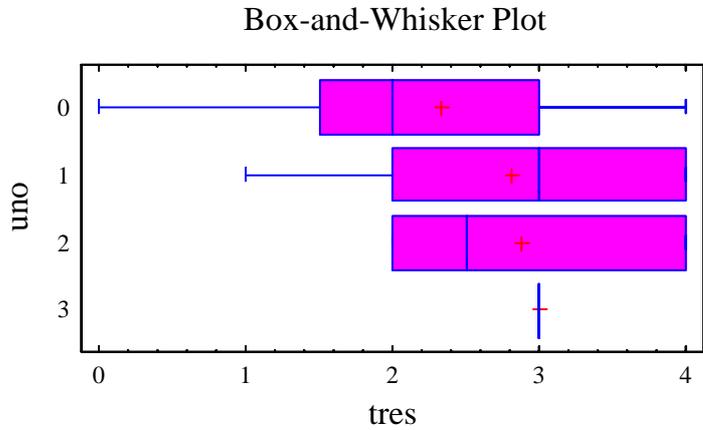


Table of Means for tres by uno with 95,0 percent LSD intervals

uno	Count	Mean	Std. error (pooled s)	Lower limit	Upper limit
0	24	2,33333	0,209675	2,03523	2,63144
1	16	2,8125	0,256798	2,4474	3,1776
2	8	2,875	0,363167	2,35867	3,39133
3	4	3,0	0,513596	2,2698	3,7302
Total	52	2,61538			

This table shows the mean for each sample. It also shows the standard error of each mean, which is a measure of its sampling variability. The standard error is formed by dividing the pooled standard deviation by the square root of the number of observations at each level. The table also displays an interval around each mean. The intervals currently displayed are based on Fisher's least significant difference (LSD) procedure. They are constructed in such a way that if two means are the same, their intervals will overlap 95,0% of the time. You can display the intervals graphically by selecting Means Plot from the list of Graphical Options. In the Multiple Range Tests, these intervals are used to determine which means are significantly different from which others.

Kruskal-Wallis Test for tres by uno

uno	Sample Size	Average Rank
0	24	22,875
1	16	28,9375
2	8	29,5
3	4	32,5

Test statistic = 2,95695 P-Value = 0,398307

Our conclusion: Students familiar with WinHipe exhibit improved comprehension of the algorithm assigned to create animations. Nevertheless, this difference is statistically (Kruskal-Wallis) rather weak.

Hypothesis 4: "The previous use of WinHIPE helps to understand the algorithm to be debugged."

Statistical analysis used: Multiple-Sample Comparison.

uno	Count	Average
0	24	1,20833
1	16	1,5625
2	8	2,125
3	4	2,0
Total	52	1,51923

uno	Minimum	Maximum
0	0,0	3,0
1	0,0	4,0
2	1,0	4,0
3	1,0	3,0
Total	0,0	4,0

Box-and-Whisker Plot

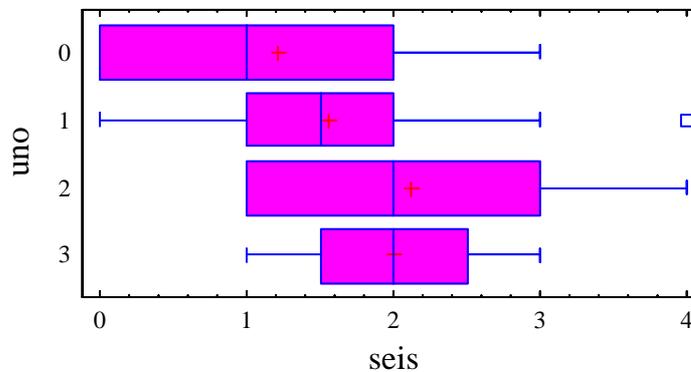


Table of Means for seis by uno with 95,0 percent LSD intervals

uno	Count	Mean	Stnd. error (pooled s)	Lower limit	Upper limit
0	24	1,20833	0,214028	0,904042	1,51262
1	16	1,5625	0,26213	1,18982	1,93518
2	8	2,125	0,370707	1,59795	2,65205
3	4	2,0	0,52426	1,25464	2,74536
Total	52	1,51923			

uno	Sample Size	Average Rank
0	24	22,5833
1	16	26,8438
2	8	33,875
3	4	33,875

Test statistic = 4,8056 P-Value = 0,186595

Our conclusion: Although this algorithm was understood worse (probably, due to its greater difficulty), students familiar with WinHipe exhibit better comprehension of the algorithm assigned to debug.

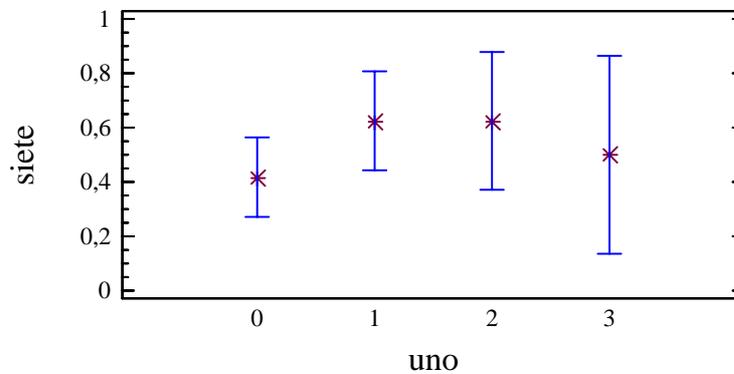
Hypothesis 5: “Students use animation capabilities to debug the algorithm, independently of their previous experience with them.”

Statistical analysis used: Multiple-Sample Comparison.

uno	Minimum	Maximum
0	0,0	1,0
1	0,0	1,0
2	0,0	1,0
3	0,0	1,0
Total	0,0	1,0

WARNING: The standardized skewness and/or kurtosis is outside the range of -2 to +2 for 2 columns. This indicates some significant nonnormality in the data, which violates the assumption that the data come from normal distributions. You may wish to transform the data or use the Kruskal-Wallis test to compare the medians instead of the means.

Means and 95,0 Percent LSD Intervals



ANOVA Table for siete by uno

Analysis of Variance					
Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Between groups	0,522436	3	0,174145	0,67	0,5740
Within groups	12,4583	48	0,259549		
Total (Corr.)	12,9808	51			

The ANOVA table decomposes the variance of the data into two components: a between-group component and a within-group component. The F-ratio, which in this case equals 0,670954, is a ratio of the between-group estimate to the within-group estimate. Since the P-value of the F-test is greater than or equal to 0,05, there is not a statistically significant difference between the means of the 4 variables at the 95,0% confidence level.

Box-and-Whisker Plot

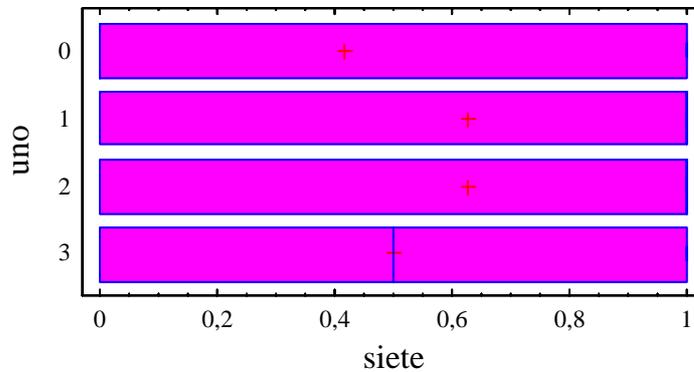


Table of Means for siete by uno with 95,0 percent LSD intervals

uno	Count	Mean	Std. error (pooled s)	Lower limit	Upper limit
0	24	0,416667	0,103993	0,268816	0,564517
1	16	0,625	0,127365	0,443921	0,806079
2	8	0,625	0,180121	0,368915	0,881085
3	4	0,5	0,25473	0,137842	0,862158
Total	52	0,519231			

Our conclusion: The use of animations by students does not depend on the previous use of WinHIPE. Therefore, using program animations with our environment is easy to learn and easy to use.

Hypothesis 6: “Students who think that animations are easy to use, use them for debugging.”

Statistical analysis used: Multiple-Sample Comparison.

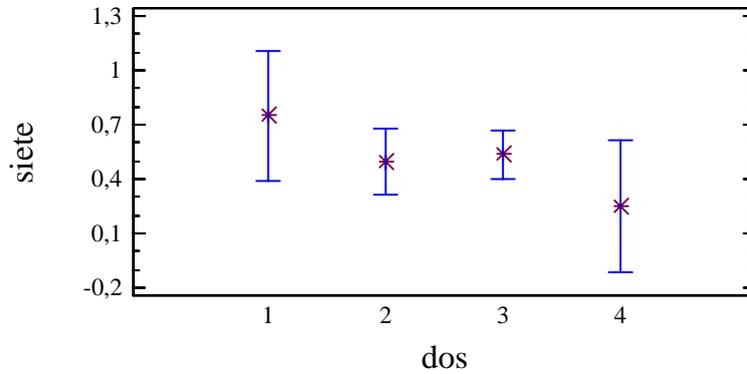
Summary Statistics for siete

dos	Count	Average
1	4	0,75
2	16	0,5
3	28	0,535714
4	4	0,25
Total	52	0,519231

dos	Minimum	Maximum
1	0,0	1,0
2	0,0	1,0
3	0,0	1,0
4	0,0	1,0
Total	0,0	1,0

WARNING: The standardized skewness and/or kurtosis is outside the range of -2 to +2 for 1 columns. This indicates some significant nonnormality in the data, which violates the assumption that the data come from normal distributions. You may wish to transform the data or use the Kruskal-Wallis test to compare the medians instead of the means.

Means and 95,0 Percent LSD Intervals



Box-and-Whisker Plot

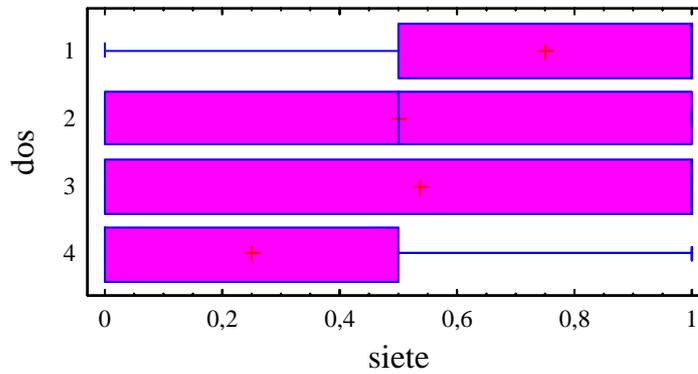


Table of Means for siete by dos with 95,0 percent LSD intervals

dos	Count	Mean	Std. error (pooled s)	Lower limit	Upper limit
1	4	0,75	0,25479	0,387755	1,11224
2	16	0,5	0,127395	0,318878	0,681122
3	28	0,535714	0,0963017	0,398799	0,67263
4	4	0,25	0,25479	-0,112245	0,612245
Total	52	0,519231			

Kruskal-Wallis Test for siete by dos

dos	Sample Size	Average Rank
1	4	32,5
2	16	26,0
3	28	26,9286
4	4	19,5

Test statistic = 2,02921 P-Value = 0,566364

Our conclusion: Animations are less used by students who think that animations are easy to use.

Appendix C: Tabulated data

Each row contains the answers of a returned questionnaire. Each column contains the answer to the corresponding question.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
2	2	3	3	2	1	1	2	3	2	3	2	2	1	4	0	3	3
0	2	1	1	2	1	0	2	3	0	3	1	2	2	0	0	3	3
1	3	1	3	1	4	1	2	1	3	3	3	2	2	2	1	3	3
0	3	4	2	1	1	1	2	3	0	2	1	2	0	3	0	2	2
0	2	2	3	2	2	0	3	2	2	2	2	1	1	1	1	2	2
0	3	3	3	2	1	0	3	3	3	3	3	2	1	3	1	3	2
0	4	0	0	0	2	0	3	1	3	3	3	1	1	1	0	2	2
2	3	2	3	3	3	1	2	3	1	3	4	2	1	2	0	3	2
0	3	3	4	2	1	1	4	2	0	3	2	3	2	0	1	3	2
1	1	2	3	2	2	1	3	3	0	3	4	3	3	2	1	3	4
1	3	3	3	2	1	0	3	3	1	4	4	3	2	2	0	3	2
2	3	4	3	2	2	1	3	2	0	2	3	1	0	0	0	2	2
1	2	2	2	0	0	1	3	2	2	3	4	2	2	2	0	2	3
1	3	3	1	2	1	0	2	1	0	3	3	3	2	0	1	4	3
0	1	1	2	2	1	1	3	1	0	3	2	3	2	1	0	3	2
2	2	2	3	2	1	1	2	3	1	1	3	0	0	0	1	2	2
2	3	4	2	2	3	0	2	3	0	2	3	2	1	3	1	2	3
1	3	4	0	0	3	1	2	3	1	2	2	2	2	2	1	3	2
1	2	2	1	1	2	1	3	2	0	3	3	3	2	2	1	4	2
1	4	4	2	2	1	1	3	2	0	2	4	1	0	2	0	1	3
0	3	2	3	2	2	0	2	2	1	2	2	3	1	0	0	1	1
1	2	2	4	3	1	1	3	2	1	3	3	3	0	1	1	3	1
0	3	2	2	1	0	0	1	2	1	4	4	1	0	3	0	2	3
0	3	4	3	1	2	0	3	2	0	3	3	4	2	2	1	3	3
0	3	2	2	1	0	1	2	2	0	1	1	1	0	0	0	1	1
3	2	3	2	1	2	0	3	3	0	2	2	0	0	1	0	1	2

References

- [1] Naharro-Berrocal, F., Pareja-Flores, C. and Velázquez-Iturbide, J.Á. “Automatic generation of algorithm animations in a programming environment”. *Proc. 30th ASEE/IEEE Frontiers in Education Conf.* (Kansas City, Missouri, USA, 2000), 6-12 (session S2C).

- [2] Naharro-Berrocal, F., Velázquez-Iturbide, J.Á., Pareja-Flores, C. and Medina-Sánchez, M.Á. “Valoración de la eficacia de las animaciones de algoritmos en el aprendizaje de la programación funcional”. Unpublished report, 2002. Available at <http://vido.escet.urjc.es/winhipe/docs/209-erpmeuiona.pdf>