

GreedEx Tab: Tool for Learning Greedy Algorithms on Mobile Devices

Manuel Ortega Cordovilla, José David Patón Romero, Miguel Ángel Redondo Duque, Manuel Ortega Cantero,
Ana Isabel Molina Díaz

College of Computer Science Engineering
University of Castilla-La Mancha (UCLM)
Ciudad Real, Spain

e-mail: Manuel.Ortega1@alu.uclm.es, JDavid.Paton@uclm.es, Miguel.Redondo@uclm.es, Manuel.Ortega@uclm.es,
AnaIsabel.Molina@uclm.es

Abstract— This paper describes the process proposed in order to develop the *GreedEx Tab* mobile application for the Apple iPad. This application will handle the technology of this device to help the user learn greedy algorithms. This addresses the problems detected in the literature on the understanding of this programming schema by students and also the issues with the former desktop application *Greedex*. Thus, the application allows a student to perform the simulation with greedy algorithms for several problems related with knapsacks. After this simulation, the state of the knapsack will be shown in three dimensions and so will the data generated as a result. Preliminary results on the use of this tool show an improvement in the performance of the students using *GreedEx Tab* over the results achieved with the previous development, named *Greedex*.

Keywords- Greedy algorithms learning; m-Learning; Knapsack problem; iOS; iPad.

I. INTRODUCTION

A. Statement of the Problem

GreedEx [17] is a Java application developed by the University Rey Juan Carlos of Madrid (Spain) within the LITE (Laboratory of Information Technologies in Education) research group, whose objective is to facilitate the learning of greedy algorithms [2].

A greedy algorithm is a strategy that works well on optimization problems with the following characteristics:

1. Greedy-choice property: A global optimum can be obtained by selecting a local optimum.
2. Optimal substructure: An optimal solution to the problem contains an optimal solution to subproblems.

This kind of algorithms are difficult for students to understand [12].

This application has been evaluated by the CHICO (Computer Human Interaction and Collaboration) research group of the University of Castilla-La Mancha (Spain), in collaboration with the LITE research group of the University Rey Juan Carlos, using an eye tracking device available in the CHICO usability laboratory of the College of Computer Science Engineering of Ciudad Real.

The results of this evaluation [5] [6] have identified a number of shortcomings in the graphical user interface and in the learning methods that suggest that a more interactive tool could improve this learning. In [6] with the aim of

improving the learning experience of students with the former Greedy system, we have carried out two experiments in order to assess the representations that this system supports. In these experiments, the information collected by means of subjective perception and motivation questionnaires was combined with the information obtained from an eye-tracker device.

For example, one of the conclusions reached was that the students preferred the tabular representations because the graphical representations were more complex and less representative, when it should be the other way around. In addition, the graphical representations lacked an interactive nature, which is an added difficulty.

Therefore, it is interesting to make an application where graphical representations take a greater role and actually help in learning greedy algorithms, improving, among other things, the interaction with the representation of the results.

B. Objectives of the Work

The main objective of this research is to implement a new version of the *GreedEx* application for iPad (*GreedEx Tab* [14]), in order to improve and solve the problems encountered in the studies conducted on it.

This new version has to take advantage of the convenience of use and the ability of interaction, visualization, and animation of mobile devices, in order that users can learn the behavior of greedy algorithms. In this way, we intend to improve the learning of greedy algorithms as well as directly and objectively measure this improvement through eye tracking techniques.

Among the mobile devices currently available, we have opted for the iPad tablet, since the technology provides the necessary resources to develop this application. In addition, the *GreedEx* application is available only for conventional computers. Therefore, it is intended to extend the learning possibilities through this tool for other devices, such as mobile devices (in this case, the iPad).

C. Target of the Application

The developed application is targeted for use in university teaching in specialties and/or degrees of computing, in the field of learning of greedy algorithms. The subjects involved have different names at different universities, but the contents are framed in the typical algorithmic schemes that all computer engineering degrees have.

Furthermore, it is expected to be used by research groups in their work or part of it is based on the study and teaching of greedy algorithms, as is the case of the research groups of CHICO of the University of Castilla-La Mancha and LITE of the University Rey Juan Carlos.

Thus, this application is expected to have some impact on these areas and its use leads to the improvement of learning and research relating to greedy algorithms, as well as emergence of new applications related to this field and improvements/updates in the application in question.

In the present course 2015/2016 this application is used in several subjects of the Studies on Computer Science at the University of Castilla-La Mancha. Preliminary results show that the application fits the requirements needed in order to understand the complexity of the greedy algorithms and test over satisfaction and eye tracking studies will measure the improvement of this application in comparison with the former application *Greedex*. Some of these preliminary results could be shown at [15].

This article is structured into the following sections: first, in Section II we present the state of the art of this work. Section III focuses on presenting the working method used to develop the work. In Section IV we explain the Results obtained from our approach. Finally, in Section V, we discuss the conclusions arising from this work and the future work derived from it.

II. STATE OF THE ART

A. Greedex

The name of the *GreedEx* application is based on the acronym “GREEDY algorithms EXperimentation”.

The *GreedEx* application is oriented to help in the learning of greedy algorithms. To this end, it gives support to interactive experimentation with different selection functions for a given problem, which is solved with a determined greedy algorithm.

The interactive experimentation on a particular problem allows the user to enter the data, execute greedy algorithm to reach a solution, and compare the obtained results.

In its current version, *Greedex* supports six optimization problems. One is the problem of selecting an activity, while the other five consist of exercises of the knapsack.

Among these exercises of the knapsack, the fractional knapsack, the 0/1 knapsack, and the variants of these exercises can be found. Each of these exercises has some selection functions with which the corresponding solution can be reached.

In Figure 1, the screen that shows *GreedEx* after solving a specific problem can be observed. In this case, it is the problem of the maximum number of objects, which is a variant of the exercise of 0/1 knapsack.

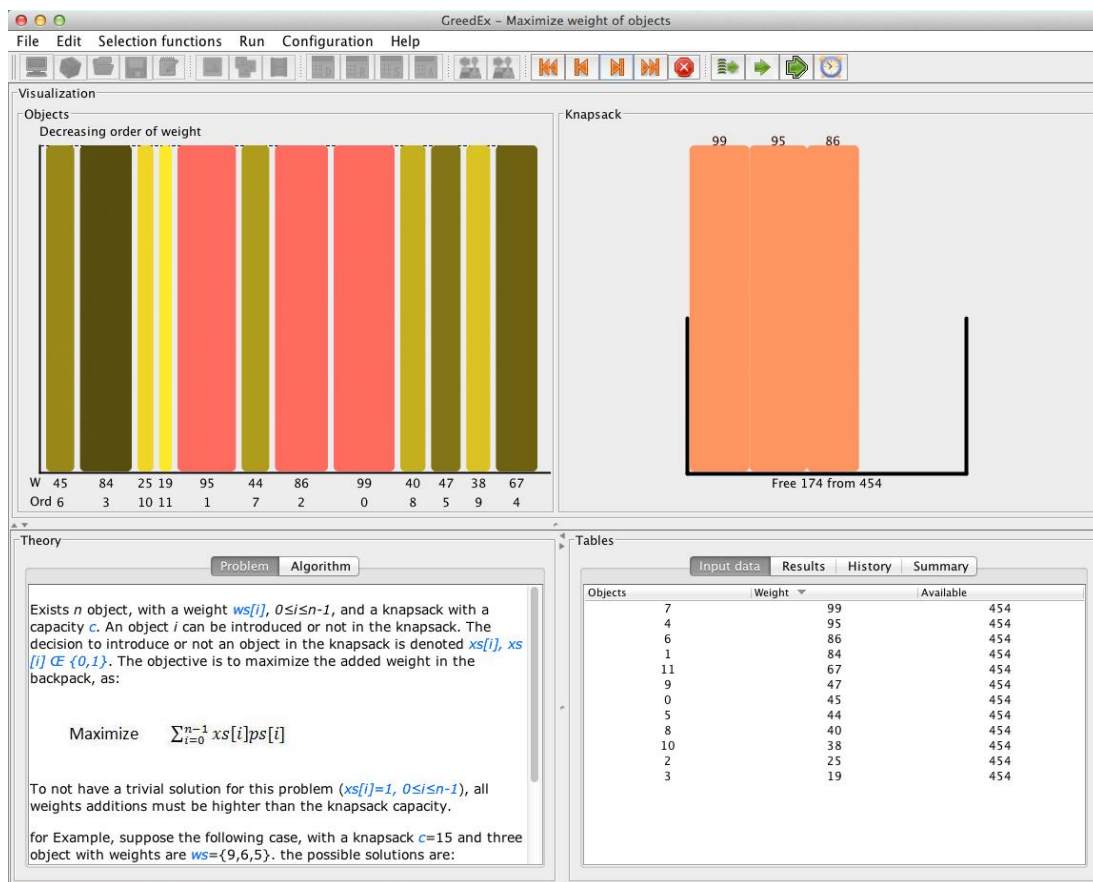


Figure 1. *GreedEx* screen after solving a specific problem with a greedy algorithm.

B. Development Technologies

Firstly, the device on which the development of the application in question has been carried out is the iPad, over iOS 9. The larger size of the device, which has a bigger screen, has been one of the motives for choosing this mobile device, since the representations to display will be seen much better and will be much more practical to orient the development, for example, for iPhone.

Secondly, to develop applications for iOS, it is necessary to use an integrated development environment (IDE). In this case, it is Xcode [18]. This is the integrated development environment of Apple, specific to developing applications for iOS and OS X.

Lastly, Objective-C as a programming language has been chosen. This is one of the major programming languages in Xcode to develop applications for iOS and OS X.

III. WORKING METHOD

A. SCRUM as Management Methodology

The research project development requires an agile management methodology, light, and, of course, flexible, due to the few people involved in it. We used a combination of OpenUP and SCRUM [10] methodologies and our methodology CIAM [4] [7] and CTTE [8] for task analysis, following a common schema in our CHICO Laboratory.

It is noteworthy to mention that SCRUM was conceived from the idea of following the *Manifesto for Agile Software Development* [1], and this has been one of the main reasons that motivated the choice of this methodology.

For our development users are teachers and students involved in the classes of subjects in Advanced Programming.

B. OpenUP as Development Methodology

As is the case with SCRUM, OpenUP [16] is also an agile methodology and was designed with the idea of following the *Manifesto for Agile Software Development* [1], and these are, again, the main reasons why OpenUP has been chosen as the development methodology.

Although sufficient and complete, OpenUP continues to be a variant of RUP (Rational Unified Process) [3]. That is why it retains some of their aspects, as is the structure of the life cycle divided into four phases (initiation, elaboration, construction and transition).

However, OpenUP refines the concepts inherited from RUP, to treat only basic aspects of project development, together with new concepts oriented toward facilitating the development of small projects.

All these reasons led us to choose OpenUP as the development methodology for the research.

C. Application of the Working Method

Based on the selected methodologies (SCRUM and OpenUP), the research was managed by *Sprints* with a duration of one week each, while the development was carried out by iterations according to a sequence of methods achieved through microincrements.

Although our aim is to do some research in order to facilitate the study of the greedy algorithms, in this paragraph we will use the term “Project” to refer to our research because these phases relate to the development of the application itself.

For the full development twelve *Sprints* were carried out, which are divided into four phases. The aspects that have been covered in each of these phases are initiation, elaboration, construction and transition.

In the transition phase, already having the final version of the application, the work was based on carrying out all necessary steps to finish the project and launch the final version of *GreedEx Tab*. Thus, a verification of the software quality was performed, performing black- and white-box testing, such as usability testing using the eye tracker with various students of the College of Computer Science Engineering of Ciudad Real, all in order to confirm if the expectations and objectives of the project had been met.

IV. RESULTS

In Table I, the features of *GreedEx Tab* can be observed.

TABLE I. FEATURES OF *GreedEx Tab* APPLICATION

<i>Functionalities</i>	
1. Allow selection between different knapsack exercises.	6. Allow access to historical simulations where the data and results thereof can be observed.
2. Allow manual or random introduction of the necessary data for simulations.	7. Having an “About” section, where information about the application is displayed, along with help about the function.
3. Allow selection of different selection functions.	8. Allow a connection with iCloud, in order to store and load data from different simulations.
4. Allow making step by step, all steps at once, and intensive simulations.	
5. Allow the state the state of the knapsack to be viewed in three dimensions after a simulation.	
<i>Graphical User Interface</i>	
1. Allow navigation between the theoretical information of an exercise of the knapsack, the data used for simulation, the simulation of the exercise, and the historical data with the results.	4. When the knapsack is represented in three dimensions, display a legend about items by which it is formed.
2. Provide the necessary mechanisms to allow the introduction of the necessary data for the simulations.	5. Provide differentiation between elements of different kinds.
3. Graphically and textually show the items in the knapsack (or knapsacks), as well as those who are still out.	6. Provide feedback from those events that occur, that are in progress, or errors that may arise.
	7. Follow the principles of Human-Computer Interaction, at no time using the application must become a tedious task.
<i>Performance</i>	
1. Perform simulations within reasonable limits of time and memory for this purpose.	

Different example views of the *GreedEx Tab* application are shown in Figures 2 to 13, and there how it is developed, how it works, its graphical interface, etc., can be observed.

A. Application Startup

At the startup of *GreedEx Tab*, the initial screen of the application is displayed, as shown in Figure 2, from which a user can start to interact with it.

From this view, we can see information about the version of the application (“Greedex Tab v1.3”), get access to the information and available help about the application (using the button “About Greedex Tab”), or select a specific knapsack exercise about which the simulation will be performed (among other features).

B. Theory of an Exercise View

Figure 3 shows the view displayed when a specific knapsack exercise is selected.

In that view, by default, the text and the code of the selected exercise, namely the theoretical part of the exercise, are displayed. The text and the code appear in different views with their own representation formats. The text appears on a laid paper, while the code is displayed in an OS X window. In this way, the essence of these parts is transmitted a little better to the user.

With this view of a knapsack exercise, in the bar at the bottom, a user can navigate to the different functionalities: “Theory”, “Data”, “Simulation”, and “History”. Each of these tabs has a symbol (in addition to the identification name in question), easily identified by the user.

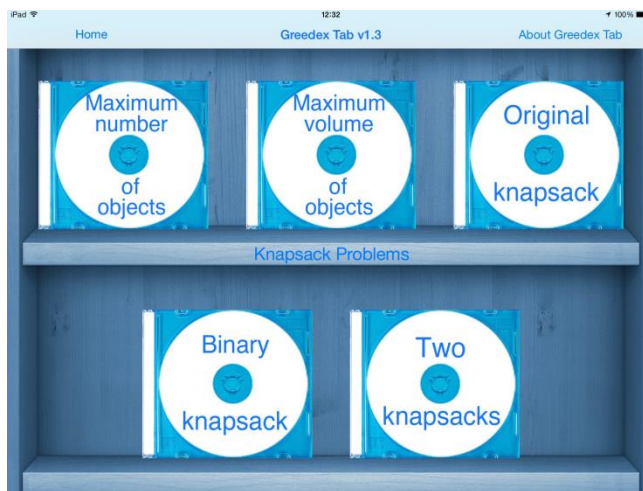


Figure 2. *GreedEx Tab* initial screen.

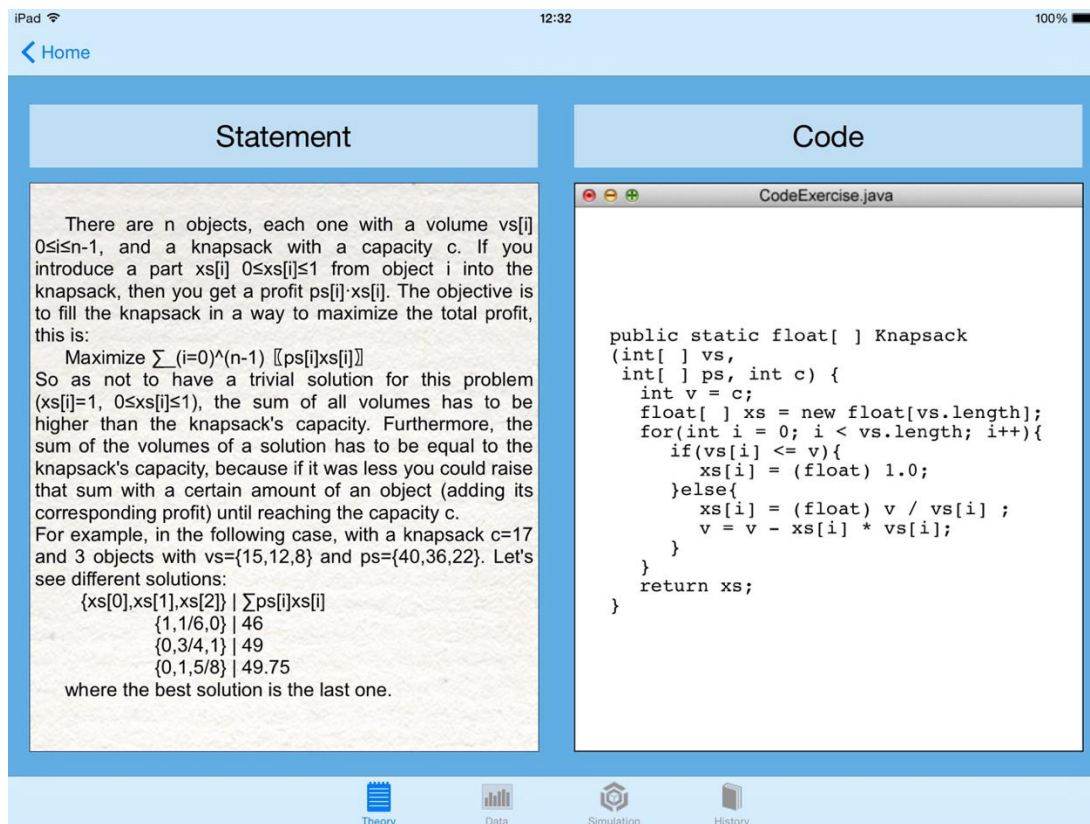


Figure 3. View of theory of an exercise of the knapsack in *GreedEx Tab*.

C. Data Creation View

Once the user has read and understood the corresponding theory of the exercise, to execute the simulation, the user must access the “Data” tab, where the data of the knapsack capacity, the maximum benefit by object, and the number of objects to create must be established, as shown in Figure 4.

In case of the exercise of two knapsacks, the view also includes a part to establish the capacity of the second knapsack.

Once the necessary data have been established, the result is the one shown in Figure 5.

Note that the data for the simulation can be entered manually or generated randomly by using the “Random data” button.

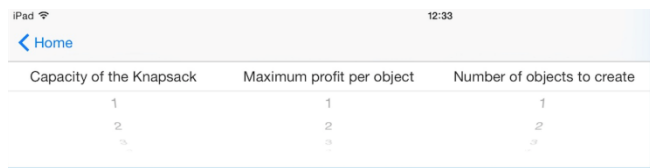


Figure 4. Part of the view dedicated to establishing the essential data for the simulation.

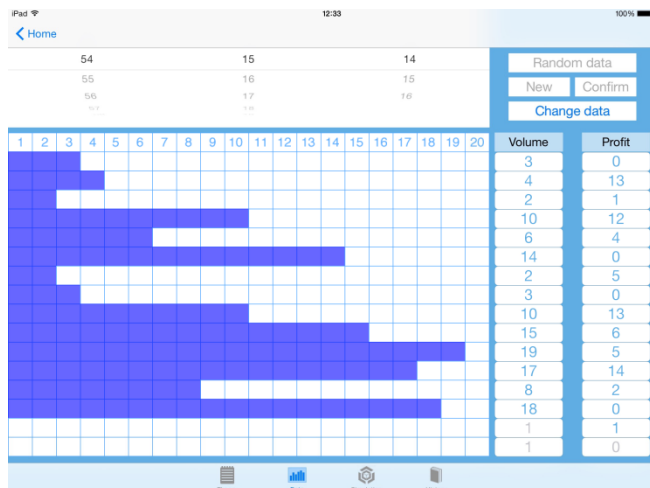


Figure 5. View with the data of the objects to create.

D. Simulation Objects View

Once the data of the objects established, these will appear created in the view of the “Simulation” tab, as shown in Figure 6.

In this view, when choosing a specific selection function, the objects created appear on the screen sorted by this selection function.

The selection functions are different depending on the knapsack exercise chosen. In Figure 7, the selection functions based on the available knapsack exercises are shown.

For example, in Figure 8, the objects created, ordered by the selection function “Increasing volume” are shown.

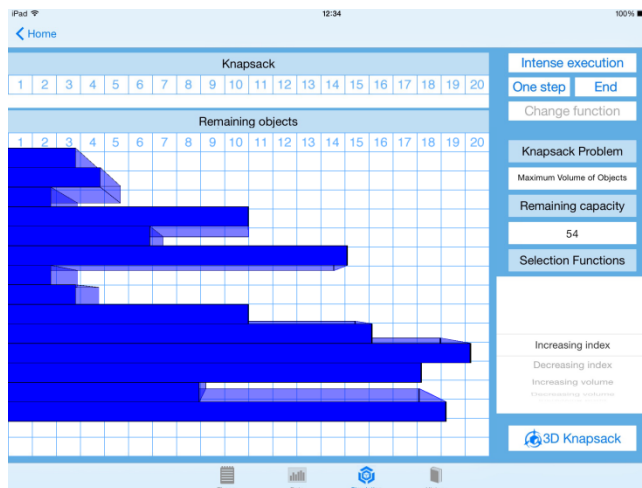


Figure 6. Simulation view with the created objects.

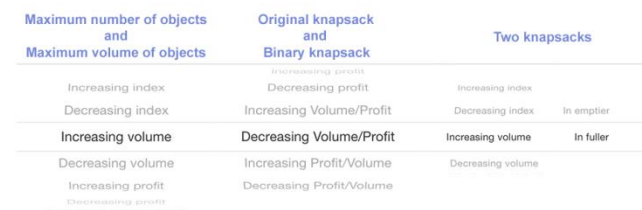


Figure 7. Selection functions according to the exercises of the knapsack.

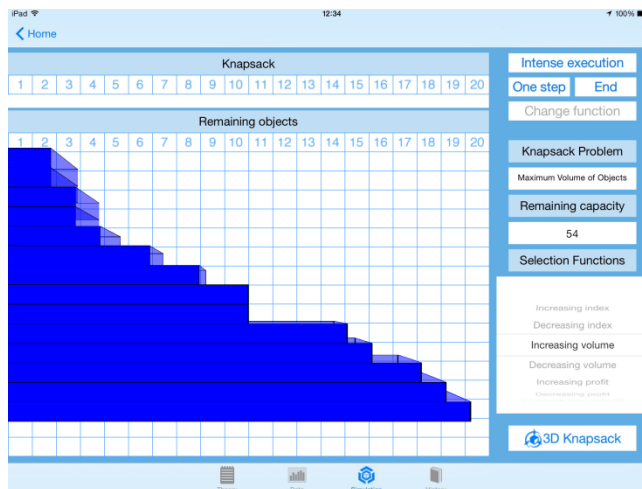


Figure 8. Sorting objects according to the selection function chosen.

To carry out the simulation that will introduce the objects in the knapsack and generate a series of results, there are three possibilities: step by step, all the steps at once, and intensive.

Firstly, using the example of sorting of the selection function chosen in Figure 8, when the step by step execution in the simulation (using the button “One step”) is performed, the first object is inserted into the knapsack, which will reduce its remaining capacity, as can be seen in Figure 9.

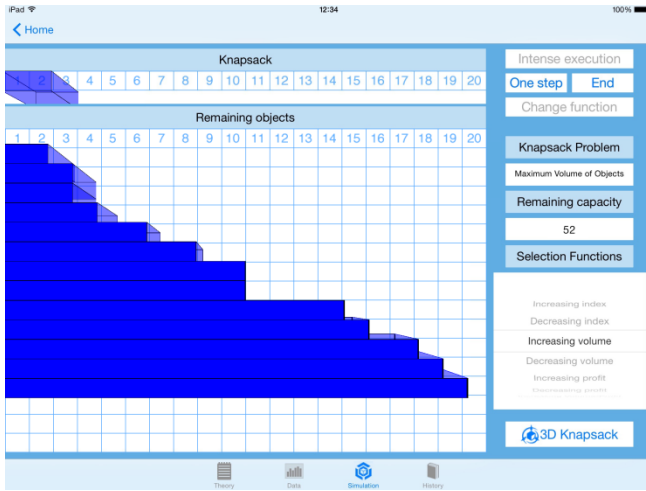


Figure 9. Introduction of an object in the knapsack when executing a step in the simulation.

Secondly, instead of carrying out the step by step execution, all the steps of a particular selection function can be performed at once, using the “Final” button, which will introduce all remaining objects following the order established, as shown in Figure 10.

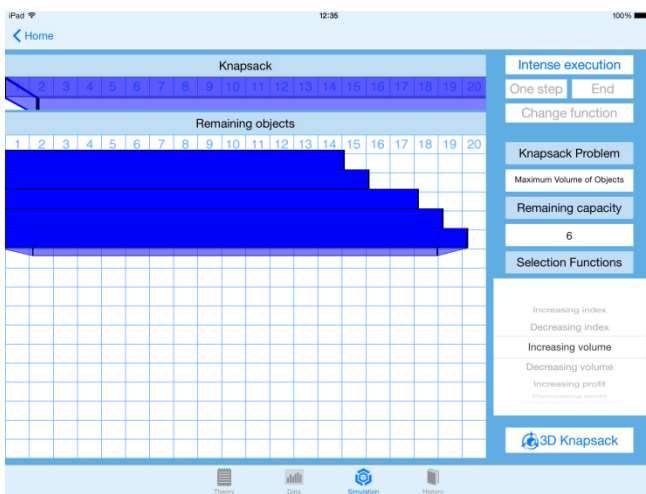


Figure 10. Final execution in the simulation that introduces all the objects in the knapsack.

Finally, the user also has the option to make an intensive execution (using the button with the same name). It is, therefore, not necessary to choose a selection function, since it is made with all the selection functions available in the exercise.

Note that the simulation view in the exercise of the two knapsacks differs from the rest because, logically, it contains two knapsacks to introduce objects and, therefore, two remaining capacities, as can be seen in Figure 11.

E. Three Dimensions Knapsack View

After carrying out the execution of “One Step” or “Final”, the user can view the status of the knapsack in three dimensions by pressing the button “Knapsack in 3D” which

can be found in the “Simulation” tab, which will display the view on the screen shown in Figure 12. The user can interact with this knapsack in three dimensions, being able to rotate it and view it in its entirety.

It is important to note that this representation of the knapsack in three dimensions has been implemented using the OpenGL graphics library [13].

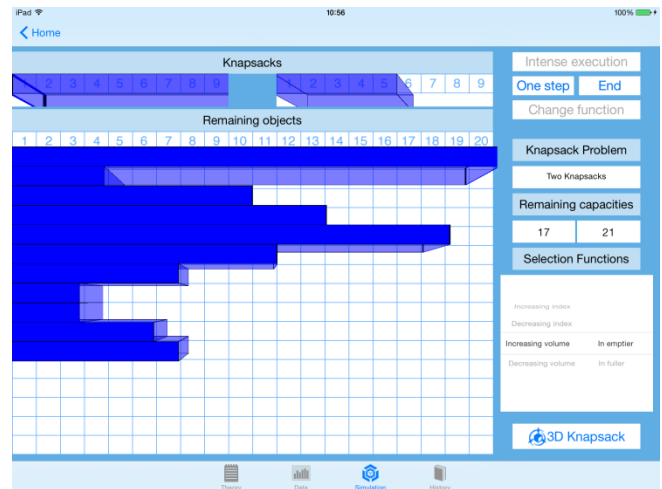


Figure 11. View of simulation with two knapsacks.

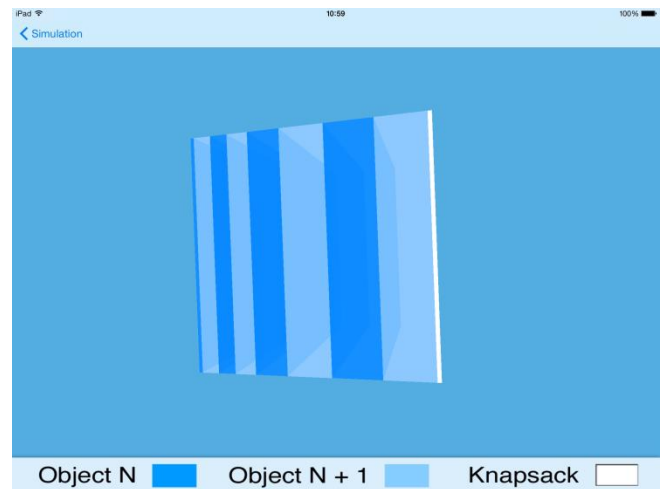


Figure 12. View of the knapsack in three dimensions.

F. Historical Data of Simulations View

All executions that are performed for a specific exercise in a session are recorded and appear in the “Table of execution” of the “History” tab along with the data and information relating to them, such as the selection function, the candidates and selected objects, the remaining capacity of the knapsack, and the result.

In this view of historical data of simulations, a functionality allows a user to store and load data from any simulation in iCloud.

Whether the data are from local executions during the current session, or loaded from iCloud, the success rates of selection functions and other affected results will be updated.

The user is notified of any access to iCloud, or whether there has been a change of data in the “Data” tab.

Figure 13 shows the view displayed in the “History” tab, in which, by way of example, the result of different executions with various selection functions is shown.

In this view, we can see the available buttons for the functionality that allows a user to upload and store the data of the executions on iCloud. These buttons, located in the upper right, are represented with their characteristic symbol and with the function that they perform.

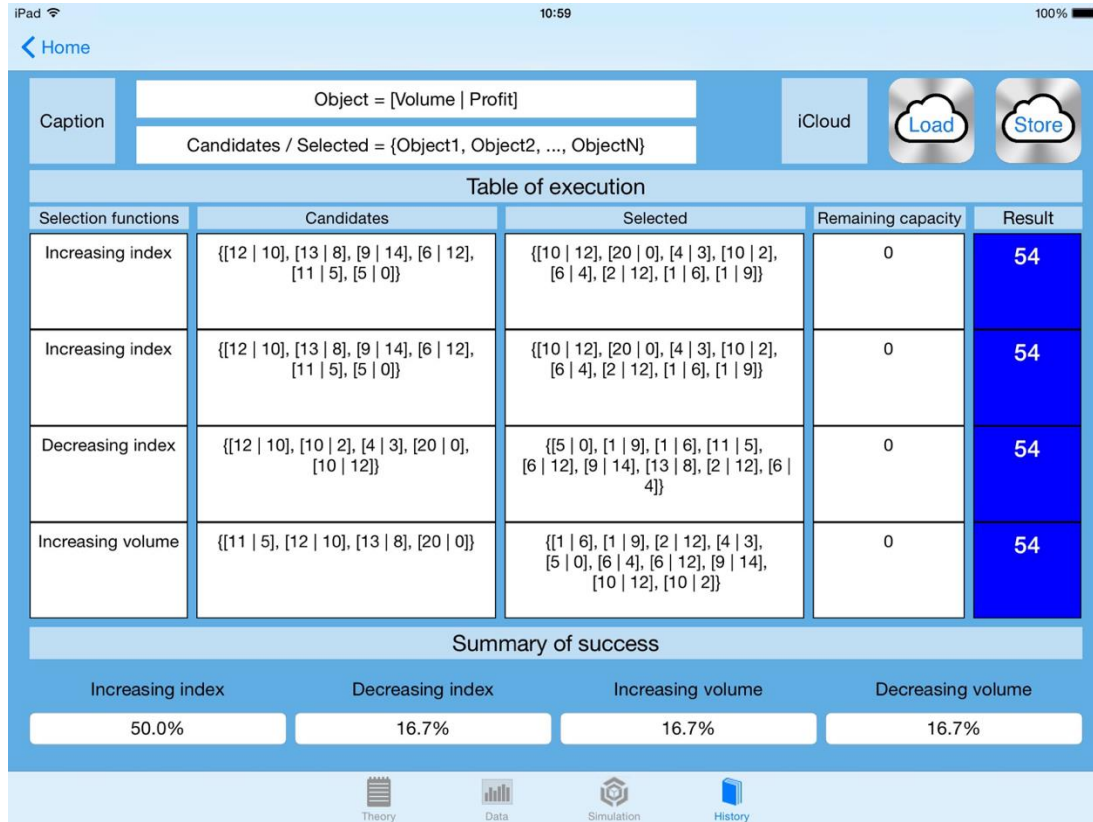


Figure 13. View of the historical data of the executions performed.

G. Internationalization

The application has full versions in English, Spanish and German, which will appear if these are the native languages installed on the iPad of the user.

H. About the Application

It is important to note that *GreedEx Tab* has been developed following the golden rules of Ben Shneiderman [11], which establishes the eight basic principles of usability in the interfaces design. In addition, the principles of designing applications for iOS established by Apple have also been followed, as well as the usability patterns for mobile devices defined by Jacob Nielsen [9].

V. CONCLUSIONS

The *GreedEx Tab* application is born with a strong didactic nature, as it is a very useful tool in learning greedy algorithms for users who are studying such algorithms.

Throughout the implementation, we have tried to make the *GreedEx Tab* an intuitive and a fast application that allows the user to interact in the simplest way possible,

giving him the functionality that he needs so that learning does not become a complicated and tedious task.

An aesthetically elegant and attractive application has been created, allowing the user to focus on functionality, characteristics that define the lines of development that Apple follows nowadays in all their applications.

A. Usability Tests

To analyze the usability improvements added to the application and check the resolution of the problems found in preliminary studies about *GreedEx*, a series of tests was carried out [15].

These tests were performed with students of the College of Computer Science Engineering of Ciudad Real, and consisted of raising a series of activities to do in the *GreedEx Tab* application with the iPad device, while registering and recording all activity with the eye tracker.

Before and after performing the tests on the application, the participants had to fill out a questionnaire about their knowledge of greedy algorithms and their experience with the application. After collecting all the information, the comparison and the analysis of the data collected was carried out through the questionnaires and the eye tracker.

After analyzing all the information, it can be determined that the shortcomings in the graphical user interface and in the learning methods detected in preliminary studies of the *GreedEx* application [5] [6] have been resolved.

The tabular representations have taken a back seat and now the graphical representations are the most consulted and most representative. Furthermore, the interaction with knapsack in three dimensions has become an attraction at the time of learning the functioning of greedy algorithms with this application.

B. Future Work Lines

Firstly, the *GreedEx Tab* application has been limited to its implementation only for iPad, but it could also be implemented for other types of mobile devices, not necessarily of the Apple brand, or also for conventional computers. The more mobile devices or computers of different brands and operating systems that can run *GreedEx Tab*, the faster the application could expand.

Therefore, a version of *GreedEx Tab* for web could be developed, to be accessible from all types of mobile devices or computers of any kind and any operating system.

Secondly, the application could be improved by increasing the available knapsack exercises, allowing for the simulation of more types of exercises.

Thirdly, considering that users of the application will belong to the field of computer science, and more specifically, they will be teachers and students from an engineering college or a professional training center, the implementation of a cooperative environment would be a marked improvement. In this environment, students and teachers could simulate knapsack exercises together and access the simulation data, so that they could discuss what the best solution is.

Furthermore, instead of each user having an individualized space on iCloud, this space could be common for all students in a class, allowing each student to contribute their efforts to better understand greedy algorithms.

Fourth, following this line, the application could also perform knowledge tests to demonstrate to the teacher that the students understand the concepts of greedy algorithms. For example, verify that the student knows how to anticipate the corresponding simulation algorithm and knows which object will be inserted into the knapsack, or, given a particular exercise of the knapsack, the student knows how to estimate which selection function will fulfil its purpose better.

Finally, *GreedEx Tab* has been developed in Spanish, English and German. Additionally, they could be implemented in more languages, such as French, Italian, etc.

ACKNOWLEDGMENTS

This research has been partially funded by the Ministry of Science and Innovation through the project TIN2015-66731-C2-2-R, the Thematic Network of CYTED 513RT0481, and the JCCM Projects PPEII-2014-012A and PPII11-0013-1219.

REFERENCES

- [1] K. Beck et al., *Manifesto for Agile Software Development*. Agile Alliance, Wasatch, UT, USA, 2001.
- [2] E. Horowitz, S. Sahni, and S. Rajasekaran, *Computer Algorithms / C++ (2nd Edition)*. Silicon Press, Summit, NJ, USA, 2007.
- [3] P. Kroll, P. Kruchten, and G. Booch, *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 2003.
- [4] A. I. Molina et al., "CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems," *Advances in Engineering Software*, Vol. 52, pp. 10-29, October 2012, doi:10.1016/j.advengsoft.2012.06.002.
- [5] A. I. Molina, M. Paredes, M. A. Redondo, A. Velázquez, and M. Ortega, "Evaluación basada en eye tracking de las técnicas de visualización de programas soportadas por el Sistema GreedEx," *SINTICE 2013*, September 2013.
- [6] A. I. Molina, M. Paredes, M. A. Redondo, and A. Velázquez, "Assessing representation techniques of programs supported by GreedEx," *The Sixteenth International Symposium on Computers in Education (SIEE 2014)*, Vol. 56-58, November 2014, doi:10.1109/SIEE.2014.7017704.
- [7] A. I. Molina, M. A. Redondo, and M. Ortega, "A methodological approach for user interface development of collaborative applications: A case study," *Science of Computer Programming*, Vol. 74, No. 9, pp. 754-776, July 2009, doi:10.1016/j.scico.2009.03.001.
- [8] G. Mori, F. Paternò, and C. Santoro, "CTTE: support for developing and analyzing task models for interactive system design," *IEEE Transactions on Software Engineering*, Vol. 28, No. 8, pp. 797-813, August 2002, doi:10.1109/TSE.2002.1027801.
- [9] J. Nielsen, and B. Raluca, *Mobile Usability*. New Riders, Berkeley, CA, USA, 2013.
- [10] K. Schwaber, "SCRUM Development Process," *OOPSLA Business Object Design and Implementation Workshop*. Advanced Development Methods Inc., Burlington, MA, USA, 1997.
- [11] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition)*. Addison-Wesley Publishing Co., Reading, MA, USA, 2010.
- [12] J. Á. Velázquez-Iturbide and A. Pérez-Carrasco, "Active learning of greedy algorithms by means of interactive experimentation," *ACM SIGCSE Bulletin – ITiCSE '09*, Vol. 41, Issue 3, pp. 119-123, September 2009, doi:10.1145/1595496.1562917.
- [13] Apple Inc., *OpenGL ES Programming Guide for iOS*. Retrieved March 01, 2016, from Apple Developer Website: https://developer.apple.com/library/ios/documentation/3DDrawing/Conceptual/OpenGLES_ProgrammingGuide
- [14] M. Ortega, Wiki of GreedEx Tab. Retrieved March 01, 2016, from CHICO Wiki: <http://chico.inf-cr.uclm.es/greedex/wiki/index.php>
- [15] M. Ortega, Usability Tests of GreedEx Tab. Retrieved March 01, 2016, from CHICO Wiki: http://chico.inf-cr.uclm.es/greedex/wiki/index.php/Pruebas_de_usabilidad
- [16] The Eclipse Foundation, *OpenUP*. Retrieved March 01, 2016, from EPF Eclipse Wiki: <http://epf.eclipse.org/wikis/openup/>
- [17] GreedEx Project Official Website. Available at: <http://www.lite.etsii.urjc.es/greedex/>
- [18] Xcode Official Website. Available at: <https://developer.apple.com/xcode/>