19

Educational software as a learning tool for primary school students

Marco Vannucci and Valentina Colla
Scuola Superiore S.Anna
Italy

1. Introduction

In the last years, thanks to the Internet and to the rapid development of technologies, multimedia applications and hypermedia based softwares became every day more important joining traditional media in the information diffusion framework. Moreover, due to their characteristics, these kind of tools resulted very suitable to gain children's (and not only children's) attention thus, by consequence, software producers, institutions and private programmers started using their potentialities for the development of instruments to be used expressly for teaching: the so-called *educational software (ES)*.
This chapter is organized as follows: in section 2 the main characteristics of educational software will be described together with a brief sketch of their story; in section 3 the characteristics of educational software will be analysed form a pedagogical point of view highlighting the advantages of the use of the involved technologies; in section 4 the main tools for the development of educational software will be discussed and finally in section 5 some case representative study of learning tool addressed to young users are described.

2. A global sight on educational software

The development of ES has been influenced by the parallel development of the hardware it was running on. In facts, one of the biggest problems related to the early years of ES consisted in the difficulty for teachers and users to maintain a software over the years due to the rapid change of technologies and to the lack of stable platforms.
The first examples of computer tools which can be defined *educational software* were produced within the military framework and were designed to be used to train personnel to perform several tasks. The flight simulator for instance was developed in the early 1940s and was used to generate aircraft on-board instruments data which were provided to unexperienced recruits; furthermore in the military environment, 30 years later, the synthetic radar data generator was used for similar training purposes. In that period the multimedia capabilities of computers were very limited and in most cases software could only interact with the users by means of simple graphics, texts and the keyboards. Moreover these military-tasks addressed software were isolated cases, developed to satisfy some

particular need without denoting any attitude to sketch out the guidelines for the conscious production of ES.
In the 1960s institutions start understanding the potentiality of educational software for civil purposes and some American universities – thanks to the availability of the first mainframe computers (with time sharing) – laid the basis of a solid theoretical and practical foundation for the creation of ES. In particular universities gave birth to a set of softwares to be used both by teachers in classrooms and students by their own concerning typical school topics such as maths or engineering. All the softwares were developed step by step and tested in the classrooms with students in order to assess their effectiveness (Mathis Johnson, 2003). The advantages of the use of such tools were discovered in that period and were exploited: computers were used to provide information, assess students work and manage the results obtained by the students during the tests. The most advanced titles – such as for instance PLATO whose screen shot is presented in figure 1 - were customizable and expandable by teachers in order to allow a virtually infinite collection of lessons and tests and embedded ad-hoc languages for the creation of new contents.

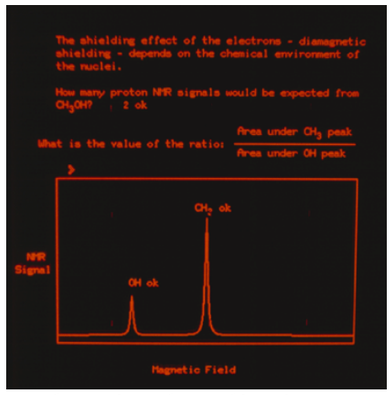

Fig. 1. A screenshot depicting an educational application running on the PLATO terminal.

The appearance of microcomputers on the market together with the experience gained in this years by software producers and by those teachers who started using computer is at the basis of the expansion of the successive years (1970s-1980s). The coming of microcomputers at accessible prices such as Commodore PET, the Apple II, the Atari XL/XE, the Sinclair which were often equipped with the BASIC programming language allowed people to create software at home or at school. In this period a quantity of programs were produced and between them some of the most creative and famous tools for students. The new ES were now designed to be captivating, to get students attention, they were at the same time challenging and simple to use and had a lot of success in classrooms. The major problem in this period was represented by the difficulty, for teachers and users who created or extended their own products, of distributing them to other people as programs written on a microcomputer usually did not work on another due to the lack of a real standard for the hardware and the programming languages.
In the '90s, due to the increasing number of software houses and to the improvement of computers features (graphics, sound, interfaces, platform standards, computer networks and storage devices such as CD-ROMs), a very high number of softwares for all possible subjects were produced. Unfortunately the growth of products number did not always

correspond to the same growth in terms of quality and innovation although very interesting ES were produced (this is the period of living books). Nevertheless ES obtained a lot of success and have been permanently included in school classes as powerful teaching tools.
In this period the world of ES is mainly influenced by the software companies and their commercial requirements which led to the origin of the so called *edutaintment*: products more oriented to their entertaining side by respect to the purely educational one.
Through the years many types of software have been developed for a number of different purposes and many genres have been experimented: some of them were represented by just a few titles and failed while others had a larger success and become standard types of ES. The main types can be classified on the basis of their teaching approach as follows:

- **Home learning.** In this category it is possible to group a wide set of titles devoted to the home learning of all types of users although particular attention is paid to children. It is often difficult to distinguish real educational software from other products where the gaming nature exceeds their educational nature. This kind of products were developed from the '90s to nowadays due to the large availability of technologies which facilitated on one hand the producers to create captivating software comparable to commercial games and on the other hand the diffusion of personal computers in every house.
- **Courseware**. The term courseware identifies all that software which is intended to be used as a support for school students including both tools expressly designed to be used by the teachers at school both products suitable to be used by students at home for improving their knowledge on specific topics. It is possible to find courseware addressed to students of all the ages: from primary school students to university students. Normally, due to their aim, these products present information and establish a sort of dialogue with students, asking questions and giving immediate feedback. Through the time such type of products have been expanded adding topics, knowledge improvements and in many cases have been organized in order to be used during entire courses and integrated in the curricula.
- **Edutainment**. The term edutaintment is used for describing those products corresponding to a mix between classical computer games and standard educational software. Edutainment reflects a commercial requirement of the public demanding for appealing tools (mainly for children) which must have an educational value and, at the same time, which results an optimal teaching solution for young children. It is possible to gather into this class all those games which have a secondary educational value but have not been created with a clear educational intent can be added. In this case the influence of software producers should be considered as many of them in the years tried to convince the public that their products had some teaching value although their gaming nature was clear.
- **Reference software.** In the first years of '90s the availability of CD-ROMs encouraged the diffusion of reference software such as dictionaries and encyclopedias. Many publisher put on market such kind of products which, in a first time, were the digital translation of already existing prints and in a second moment were enriched by means of multimedia features such a sounds and compressed video. Reference software obtained a large success on the market and many topic-specific products (music, cinema, history encyclopedias, dictionaries for translations...) were created and commercialized. Together with multimedia

contents the main aspects of these software was the possibility for the user to navigate through the contents of the encyclopedias by means of links and hyperlinks in the pre-Internet era. Nowadays reference software is migrating from CD-ROMs to the web and is becoming free and open-source thanks to projects such as the most famous Wikipedia which allows the user to examine on line contents and possibly to modify it.

3. The benefits of using educational software

Multimedia contents and interaction with the user are the main strong point of educational software and the features that mainly differentiate them from traditional media such books and television. Multimedia contents give the software the capability of attracting the user and children specially: cartoons, characters, dialogues, sounds and movies are able to stimulate the curiosity of younger users and push them to the exploration of the software. Moreover multimedia contents represent an upgrade of the learned topics, in facts while studying the first moon landing in 1969, users can look at the corresponding recording or while studying Mozart they can listen to his major compositions.

A high level of interaction with the software gives the user the freedom for deciding what to do: users can choose the direction to take for exploring the software and can establish the level of detail of the provided information and stop whenever they want. Links and hyper links through the educational tools invite the user to discover new scenarios, new arguments and surprising points of view supporting the development of associative thinking and of a multidimensional (and multimedial) vision of the same problem.

All these features can drastically modify the way of learning of the users changing a passive approach into the wish of learning. Instructors often claim that children learn by doing, thus certain softwares which are expressly designed to involve them actively in the learning process and giving them the control of their education seem to be a key tool for improving learning efficiency (Castro Sanchez et al., 2009). Educational software in facts are not conceived to maximize the degree of knowledge of the users on certain topics but their primary aim is to improve their *learning* (Guzdial et al., 1997) and, of course, a deep knowledge of the topics handled by the software will be an worthwhile side effect. During computer session children have the opportunity to experience by their own the search for the topics to learn, in a completely different way by respect to what happens with traditional lessons where the teacher decides for everybody. By consequence children experience their own ways of learning and thinking and become conscious of their peculiar learning mechanisms. Children will start controlling their own intellectual activities and will exploit this new skill it in their future studies.

Major software products integrate systems for the communication between users and between students and teachers. This aspect results important as it allows the students to share doubts and experiences in a natural way and they can undertake the software exploration together. Thanks to the recent development of Internet technologies the collective use of teaching tools can be done both within the same class with more students sharing the same personal computer and by students located in different geographical areas. Common techniques for the communication among users include instant messaging systems, dedicated email, voice and video call and many others which are currently evolving.

These kind of communication facilities allow the involvement of the teacher (or possibly more than one teacher) as well. Teachers can take part to the determination of the teaching paths of his students by deciding for instance what to propose them or the level of difficulty, help the students, answer their questions in real time. Moreover teachers interacting with their students can evaluate their level of knowledge and plan the subsequent teaching strategies.

A further aspect to be taken into consideration is the benefit deriving from the early use of personal computer by children. Computers are everyday more important in our society and we have to use them for many real life tasks: for hiring a DVD or to buy things on-line. Moreover an efficient use of computer and common softwares skills are required by companies for obtaining good jobs and computers will be in many cases necessary instruments in the working places. For these reasons computer basics should be taught to children as soon as possible in order not to left them behind in a very competitive high-tech world.

Another benefit provided by educational packages is the possibility for the students to self assess their learning. Normally self-assessment tests can be found within the packages at the end of each module (if software is organized in modules) or at the end of a course, so the student can verify his learning by means of quizzes and questions and evaluate his degree of knowledge of the presented topic. Students are often attracted by the test proposed by these tools due to their playful nature or to their simplicity and, by respect to traditional tests proposed by teachers in classrooms, they avoid any subjection to the reaction of the teacher or of the rest of the class.

Educational software offers several advantages directly to teachers as well. Main products are offered together with authoring software which allows to modify, extend and update both the contents of the software and the test that can be proposed for students evaluation. In literature it is possible to find many works where educational software are used in the classrooms and the taught topics are verified within the class by means of several test in order to assess the efficiency of the employed tools: in most cases (Juan Ye & Zhiyong Peng, 2004) the results obtained by educational software are encouraging as they improve users learning and are positively accepted by students who, in addition, get more interested on the handled arguments than by attending traditional lectures.

The use of educational tools moreover improves, for the teachers, the possibility of tracking students learning and advancements and can drastically reduce the time devoted to stand-up lessons giving to the instructor more time for tutoring activities (Carver, 1996).

Due to all these features, educational software gained an important role in children education, at school and at home and over the years many themes have been taught by means of such tools exploiting their exceptional multimedial capabilities: history, maths, geometry, computer science, literature, music. Beside these classical themes, further arguments have been handled by other titles in specific fields. In many cases, due to the nature of the treated topics, these latter titles were designed and addressed to older children or to adults but there are some educational tools handling with complex topics to be proposed to young children. These titles fully exploit the potential of educational software to teach in a natural and simple way concepts which could be hard to be understood by young students.

4. Tools and methods for the development of educational software

The development of educational software is a difficult task as it involves many interacting factors and the cooperation of several professional figures. The development process must in facts take into consideration both pedagogical and computer science factors which are normally handled by two distinct people or groups. The spectrum of activities carried out for the creation of educational tools is quite wide and includes software engineering, programming and the determination and implementation of the educational contents (Reis, 2007). The task moreover is made more complex due to the lack of a common framework for this purpose (Ivan & Josue, 2007). It is in effect quite hard to establish a standard for the realization of educational software because if on one hand the use of standard methods and tools is desirable for speeding up the realization phase of new products and for their flexibility, on the other hand such framework should be able to lead to the creation of deeply different products due to the extremely heterogeneous variety of educational softwares in terms of themes, audience, learning approaches and purposes.
Despite the lack of common frameworks there are some properties and requirements which are considered as very important for the success and the learning effectiveness of educational tools. Such requirements concern both technical and pedagogical aspects of the products and can be summarized as follows:

- **efficient interface**: the interface of the product and the interaction with the user must be clear and intuitive. Users – especially if children – must be able to identify the controls on the screen and understand immediately the operations related to each of them. It must be possible to move from one scene of the software to another by means of buttons easy to locate and to interpret and the objects on the screen it is possible to interact with must be highlighted. The choice of images and animations is important as well: they should be presented in a a pertinent way avoiding redundancies. The use of background music and sound effects is normally proper whilst spoken should be used only when necessary and in particular circumstances. Finally the use of *hot words* (words highlighted either for their relevance or because they are link to other pages or have particular functions) is recommended.
- **quality multimedia contents:** multimedia is one of the main aspects which differentiate educational tools from books. Although multimedia gives a fundamental contribution to the effectiveness of the software the abuse of multimedia contents is detrimental as it can distract the users from the central topics handled. The quality both in terms of pedagogical value and audio-visual grade is very important as well as it can determine the attention degree paid by the user to the proposed contents.
- **appropriate contents:** the proposed contents must be correct and presented in a suitable way by respect to the audience they are addressed to. The software moreover should be able to adapt its contents to the users, addressing to audiences of different age, goals and learning styles (Van Dam, 2005)
- **multidisciplinarity approach and contents extensions:** a multi-disciplinary nature and the presence of contents extensions is a positive feature for educational software as they make it able to attract more the interest of the users; moreover the contents extensions allow the users to choose the topics they want to deepen.

- **Modularity**: modularity allows the developers to easily change or extend the contents of the produced software and lengthen this way its life; moreover, if the software includes authoring tools, it allows the teachers to easily carry out changes for adjusting the product to users needs.
- **Portability**: educational tools must be usable on different kinds of computers, independently from their operative system and their architecture as they will be used both at school and at home. For these reason the use of programming languages and developing tools allowing a *cross-platform* use is preferable together with the capability of using the software on-line, possibly embedded in HTML web pages.

In the past educational softwares were constituted by monolithic applications developed within a software company, mainly by computer scientists and technicians resulting in products oriented to the computational efficiency rather than to pedagogical contents. This behaviour was mainly due to the market influence which in the '90s led software houses to the creation of appealing products suitable to a wide range of users fascinated by the emerging multimedia technologies.
Nowadays the educational software producers are oriented to the creation of components and libraries for a modular development of educational tools. The so created products are everyday more standardized, powerful and simple to use and present several strong points which are determining their success. Components give the possibility of providing to the *meta-users* (those people who will create the final educational tool by means of these components) some basic instruments which will constitute the basis of the final product which will be customized in the direction of particular requirements with a considerable saving in terms of time and programming efforts. The use of components allows non expert programmers to implement the software thus teachers and educators will be able to accomplish this task which was otherwise a programmers prerogative. This way teachers, instructors and topic experts will take care of the contents and the cognitive aspects and, on the other side, programmers will direct their efforts to the creation of good components with more interactivity and multimedia.
The development process of educational software through standard components follows the traditional scheme depicted in figure 2 which starts with a requirements analysis phase where contents providers are involved for establishing "*what*" is going to be developed and "*how*", then the components to be realized are determined and defined. Once this planning phases are concluded, components are implemented and in the last phase are assembled into the final software which is at last tested.

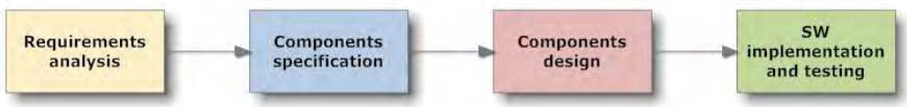

Fig. 2. Flowsheet describing the development process of educational software by means of standard components.

The use of customizable components improves both the technical and contents quality as it enables instructors and programmers to contribute to the product only in their specific fields

of interests and encourages the development of flexible and easy updatable software which will be modified – depending on the contingent necessity - by topic experts or computer scientists.
Various kinds of components are created by programmers for a multitude of functions: graphics, animation, simulation, interaction with the users, sound effects and music... Within this framework educators will add the contents by creating knowledge repository which will interact with *technical* components.
Once components are completed the educational software can be released. Normally this last phase consists of the compilation of the meta-software into the final usable product (Roschelle et al., 1999).
Softwares created by means of standardized components present some advantages for the end users as well. Users in facts will easily get used to a standard interface which will be similar for any software they are using (provided it was realized by means of the same package) and will have immediately familiarity with interactions and way of use. Furthermore the simplicity of use of the components will allow users to modify them for creating their own learning tools perfectly fitting for their learning styles and teachers will model the tools they use on the basis of classes demands.
Commercial (and not) common components can easily be obtained. Many of them are realized by means of popular programming languages such as Java. These languages allow a modular realization of components and an easy integration among technical and pedagogical issues together with a high flexibility and updating capabilities.
Components developed in Java in particular take advantage of all the capabilities of this language, creating components exploiting heredity and abstract implementations which make easy and fast the creation of the final product. Many educational softwares which can be found nowadays (often on line) have been produced by employing Java and Flash. These technologies allow the programmer to embed the final product in web pages by means of Java Applets which can be integrated in HTML pages with a few commands. For these reasons such tools are everyday more popular and users are becoming very familiar with interfaces and commands utilized in these softwares.

5. Educational software case studies

Educational software is becoming everyday more important and is becoming an integral part of educational tools used by students of all the ages. In some countries the use of such instruments has been included in curricula and teachers are involved in the development phase of many of them. Schools and institutions are favourable to the introduction of educational software and actively operate for their diffusion. In particular it can be mentioned the work carried out by the UK institution Schoolzone which, with the contribution and help of many teachers, edited an inventory of educational software on the market in order to help schools (and teachers) in the choice of software to purchase. The list – which can be found on *http://www.schoolzone.co.uk/resources/evaluations/GuideEdition6.pdf* - includes for each title a description of the software and its evaluation on the basis of several criteria.
The advantages of using educational software at school are manifold. They are acknowledged to extend the exploration capabilities of students thanks to the variety of media involved and to the quantity of information they can provide.

The appeal offered by these software makes learning more natural especially for young students so that they are spontaneously pulled toward the search of new information, playing some games. The use of educational software at school in facts contemplate as well home sessions for the students.
Simulation softwares provide a link between theory and practice so that those concepts which are taught by the teacher in the classrooms can be empirically tested by means of the computer encouraging active discovery behaviour (Morsi & Jackson, 2007). Educational tools at school are used for evaluating students preparation as well: some software – expressly designed for the use at school – include modules for the assessment of students learning and many times these evaluation are provided autonomously by the software in a transparent way by respect to the user whose feedbacks and/or answers are passed to the teacher. Teacher – beside the merely evaluation of student comprehension of topics - uses the information provided by softwares for scheduling what to teach and how to teach.
Teachers use educational software for reducing the time spent teaching in classrooms and possibly make use of this time for tutoring activities.

5.1 Educational tools: some sample case studies

Many are the themes handled by means of learning tools have been included in coursewares; most common topics are mathematics, computer science, history, literature . In the following some softwares of particular interest exploiting some of the functionalities described in this work are presented as examples.

A software for improving mathematical skills in a natural way (Polard & Duke, 2002). Pollard and Duke presented in their work an educational tool for teaching mathematics to young children by exploiting all the functionalities offered by multimedia and PC-user interactions (Wills, 1994). They noticed that although many tools for teaching basic maths exist only few of them were capable to exploit such functionalities thus they created a software which can teach simple skills. The novelty proposed by the application consists of the freedom left to the student of solving the problem with his own strategy, in facts while most maths software simply ask some questions or pose problems and wait for an answer to be entered by the user by means of a textbox, in this case students operate freely interacting with the interface which allows several solving behaviours which were previously established from the analysis of the behaviour of children using paper and pen. Five possible behaviours were analysed and implemented through five interfaces which are proposed to the users in a transparent manner.
In their work the authors show, by means of a series of test on children, that effectively users when coping with the proposed problem apply their own techniques for reaching a solution and improve their mathematical skill.
The most interesting aspects of this work are the importance acknowledged to the possibility given to users to interact in different ways with the software. This consideration lead to the design of an adaptive interface which allows users to choose their strategies.

A software for exploring art and history of Venice. VeGame (Bellotti et al., 2003) is a software developed by some researchers of Genoa University with the aim of making the visit of Venice a challenging and pleasant experience enhancing this way the quantity and

quality of information obtained by such activity. VeGame (Venice Game) is substantially a game to be run on hand-held computers which guides through the visit of most important art and history sites of the city in the context of a treasure hunt. The game – addressed to young people – can be played in teams or individually and consists of a series of mini games which require the visit of main interest places of the city (squares, churches, monuments...). Some of these games ask for the completion of inscription which can be found on monuments, some others the reconstruction of images of paintings to be observed in churches or museums, further games need the exploration of some zones for answering some questions exploiting clues present in the streets.
VeGame exploit graphics and multimedia and provides a lot of information on the city; it is possible for instance to see the map of Venice or to get information on a church the user is visiting or a painting hei is looking at. The interaction with the software is extremely simple and natural and exploits the hand-held features.
From a pedagogical point of view the software stimulates a constructive learning in the users which interact with the environment and are let to the observation of the city and to the gathering of many information for solving the riddles. The gaming nature of the software is very important as well, in facts it challenges the users which spontaneously look for information, observe and experience the exploration as a game.
From a technical point of view the software was realized in an XML based language which allows an easy integration of multimedia contents and can be simply extended and modified in the future seconding users and instructors exigences.

A software suite for the sensitization of children to environmental themes. "Three little pigs discovering materials" (Vannucci et al., 2009) is a software suite for the sensitization of primary school students to environmental themes such as the correct usage of materials, energy saving and recycling. Teaching in this case is made by relating, in the form of a highly interactive living book, the fairy tale of *The three little pigs* opportunely modified in order to relate the desired concepts.
This software is a practical example of several methodologies and features described in this chapter and employed for the realization of a tool for teaching relatively complex concepts and addressed to young users. In particular due to the age of the it was paid attention to aspects such as the active and continuous involvement of the users in order to maintain higher as possible their attention to the handled topics according to the considerations on the development of efficient teaching tools proposed by (Barbosa & Maldonado, 2006). For this reason the structure used for the development of the software follows the scheme of cartoons and comics, multimedia was used to create captivating scenarios and characters. Some games were added and entertaining elements were used in all the components of the suite for the same aim.
The suite is composed by three interconnected parts that the user can freely explore and which are listed below and subsequently described more in detail:

- **the main story:** in this part of the suite the main story of the three little pigs is told by means of interactive animations. Beside the story, many informations concerning the materials employed by the little pigs are passed to the user.
- **the knowledge improvements:** where main informations about materials, their use and their recycling process are reported together with simple quizzes and other curiosities on the treated themes.

- **the games:** some well-known games have been realized in order to verify in a funny way the learning of the concepts and to entertain the young users.

The main story by means of 35 scenes relates the tales of *The three little pigs* which is used as a pretext to discuss about the desired environmental issues such as the materials employed for building pigs houses or to perform common works. In addition to these scenes some others such as the one shown in figure 3 have been added for focusing on some material themes. During the story it is possible to interact with characters and to move from one scene to another.
Scenes were designed to be appealing to children. In particular they are colourful, full of animations and interactions. Scenes include music and sound effects and both narration and dialogues are written (by means of comics-style balloons) and spoken for those children who still can not read. Currently an Italian and English version of the software exist.

Fig. 3. Sample screen shot showing one scene used for focusing on material themes.

The main story works as a starting point for reaching additional scenes where environmental themes are handled more in detail. In particular from main story scenes it is possible to obtain further explanations on materials by clicking on some highlighted objects in the scene. For instance in figure 4 by clicking on the wooden house the user will jump to a set of scenes describing wood properties. For each of the handled materials (steel, wood, concrete, glass, plastic) several aspects are described: their story, their use and features and their recycling.
This part of the suite is characterized by the presence of many interactions with the scene elements. By clicking on objects and characters for instance the user gets explanations on particular stuffs or curious information and hints. During the narration children are asked to answer questions and play simple games on the described topics in order to keep them interested on the themes and to verify users learning.

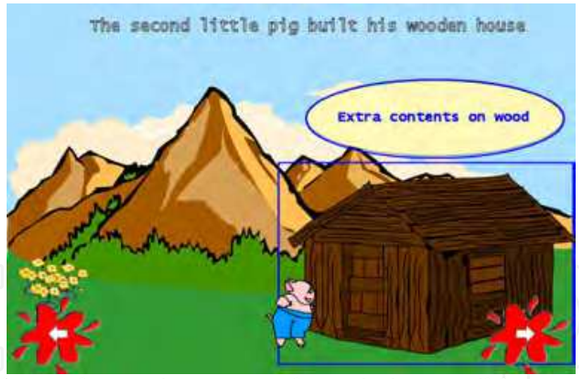

Fig. 4. Screen shots from the main story where it is possible to see the hyperlink to the special contents scenes.

The aim of the previously described parts of the suite is to provide to the users information for a correct usage of materials. In order to evaluate the comprehension of the taught concepts and to entertain the users, two interactive games were developed. The developed games maintain a link with the other parts of the suite, in facts they share characters and scenes of the main story and the same philosophy. Games were designed to be stimulating so as to favour their exploration and to make the entertaining nature emerge hiding the educational one.
The two games (whose screen shots are depicted in figure 5a and 5b) which have been implemented are very popular among youngsters and easy to be played by primary school children. The first one is the Memory game where a set of card pairs are placed on the table and the player has to find all the couples by disclosing the cards two by two. The suite includes a particular extension of this popular game: the associative memory in which couples are associated by a particular criterion (i.e. objects made on the same material) which was particularly appreciated by children.

Fig. 5. a, 5. b Screen shots from the two games included in the suite. Figure 5. a (on the left) shows the puzzle game while figure 5. b (on the right) the memory game.

The suite was implemented by using Flash animations and the Java programming language. Both these standards allowed the use of a lot of multimedia contents and, by using the ActionScript extension, the use of components characterized by high interactivity with the user which made the tool easy to use even for young children. Furthermore these technologies are familiar to children as many web pages already use it.

The suite can be either part of a stand alone executable software or embedded in an HTML page through a an Applet thanks to the emploied development tool. Flash and the Java programming language in facts grant the portability of the software which is a key issue for a wide use of the software (Beynon & Mackay, 1993).
The whole suite has been tested by a group of primary school children who interacted with the software in all its aspects and enjoyed it. In particular students liked the story and the extra contents for their cartoons-like appearance and for the interaction possibilities. Subsequently in the talks and interviews with the teachers they proved a good understanding of the key concepts handled by the software proving this way its efficiency.

6. Conclusions

In this chapter the role and the significance of educational softwares as learning tools was discussed. In particular their features were highlighted together with the advantages provided to the users (both learners and teachers) by the use of multimedia and interaction with the user which are offered by the last generations of these tools.
The evolution of educational software has been outlined drawing attention to technological development which since the first appearance of learning tools played a fundamental role in their characterization and efficiency. The key features of educational software have been pointed out and their importances have been analysed both from a technical and from a pedagogical point of view and most importance technologies for the implementation of such instruments have been described and discussed. Finally three representative case studies of existing educational tools have been described in order to show how the discussed techniques are put into practice.
From the analysis and considerations carried out in this chapter the importance of educational software emerges as they can offer to users and in particular to children a new and exciting approach to the learning process encouraging autonomy, self consciousness, a personal learning styles, the involvement of experience and a multi-sensory learning.

7. References

Barbosa, E.F.; Maldonado, J.C. Toward the establishment of a standard process for developing educational modules. Frontiers in education conference 36th annual, 2006.

Bellotti, F.; Berta, R.; De Gloria, A.; Margarine, M. VeGame: exploring art and history in Venice. IEEE Computer, September 2003.

Beynon, J.; Mackay, H. Computers into classrooms: more questions than answers. The Falmer press, London UK, 1993.

Carver, R. Computer assisted instruction for a first course in computer science. Frontiers in education conference, vol. 2, 1996.

Castro-Sanchez, J.; Del Castillo, E.; Hortolano, J.; Rodriguez, A. Designino and using software for educational purposes: FLAT, a case study. Education, IEEE transactions on, vol. 52, 2009.

Guzdial, M.; McCracken, M.; Ellito, A.. LCD: a learner centered approach to developing educational software. Frontiers in education conference proceedings, vol. 2, 1997.

Ivan, G.; Josue, G. Implementing virtual practices using an alternative methodology to develop educational software. Systems and networks communications, International conference on, 2007.

Juan Ye; Zhiyong Peng. A Squeak-based educational software – Happy learning. Creating, connecting and collaborating through computing, Second international conference on, vol. 1, 2004.

Mathis Johnson, J. (2003). From lofty beginnings to the age of accountability. A look to the past 30 years of educational software. In Learning and leading with technology, vol. 30 n.7, 2003

Morsi, R.; Jackson, E. Playing and learning? Educational gaming for engineering education. In Frontiers of education conference and global engineering, FIE2007, 2007.

Pollard, J.; Duke, R. A software design process to facilitate the teaching of mathematics. Proceedings of the international conference on computers in education (ICCE'02), 2002.

Reis, R.M. Development of educational software. Int. Journal of education and information technologies, Issue3, vol. 1, 2007.

Roschelle, J. ; DiGiano, C.; Koutlis, M.; Repenning, A.; Phillips, J.; Jackiw, N.; Suthers, D.; Developing educational software components. IEEE Computer, vol. 32, n. 9, 1999.

Van Dam, A. Visualization research problems in next generation educational software. Computer graphics and applications, IEEE, vol. 25, n. 5, 2005.

Vannucci, M.; Colla, V.; Bendinelli, L.; Colombini, M.; Massei, R.; Ciulli, S. Three little pigs discovering materials. A software for the sensitization of primari school children to environmental themes. International conference on computer supported education (CSEDU2009), Lisbon, 2009.

Will, S. Beyond browsing: making interactive multimedia interactive. Rethinking the role of education in the technology age, EdTech94 conference, Singapore, 1994.

New Achievements in Technology Education and Development
Edited by Safeeullah Soomro

ISBN 978-953-307-066-7
Hard cover, 460 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

Since many decades Education Science and Technology has an achieved tremendous recognition and has been applied to variety of disciplines, mainly Curriculum development, methodology to develop e-learning systems and education management. Many efforts have been taken to improve knowledge of students, researchers, educationists in the field of computer science and engineering. Still many problems to increase their knowledge on daily basis so this book provides newly innovations and ideas in the field of computer science and engineering to face the new challenges of current and future centuries. Basically this book open platform for creative discussion for future and current technologies to adapt new challenges in education sector at different levels which are essential to understand for the students, researchers, academic personals and industry related people to enhance their capabilities to capture new ideas and provides valuable contribution to an international community.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Marco Vannucci and Valentina Colla (2010). Educational Software as a Learning Tool for Primary School Students, New Achievements in Technology Education and Development, Safeeullah Soomro (Ed.), ISBN: 978-953-307-066-7, InTech, Available from: http://www.intechopen.com/books/new-achievements-in-technology-education-and-development/educational-software-as-a-learning-tool-for-primary-school-students

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen