The impact of interactive software on teaching

Michal Bartošovič, Viola Tamášová
DTI University, Dubnica nad Váhom, Slovakia
bartosovic@dti.sk

Abstract: *In this scientific paper, we investigate the impact of interactive software on teaching a specialized subject at a vocational school based on research conducted. Interactivity is now one of the fundamental components of ICT and digital education. Interactive software can be useful for teaching specialized subjects because it allows students to interact with the material and gain practical experience, which can enhance their learning. In the 21st century, students are increasingly identifying with the use of ICT in education, and it is necessary for teachers to continually educate themselves in their use and incorporate them into their lessons.*
Keywords*: TEACHING, EDUCATIONAL, INTERACTIVITY, ICT, SOFTWARE*

1. Introduction

We live in a time when the couch and television have become the most popular members of households, and the mobile phone is an inseparable accessory of our purses and pockets. We spend several hours a day "chilling" on the couch and surfing, "Facebooking," and playing games on our phones. In such an environment, it is very difficult to engage our children in learning through books and notebooks. Therefore, it is evident that modern techniques are being implemented and used in education, which leads to its modernization. Proper preparation and use of interactive educational materials and didactic applications is not an easy task. In many cases, it is the educator themselves who create these materials and must first select suitable educational material for processing into an interactive format, determine the amount of data to be used, and design it in a way that supports the clarity and illustrative nature of the material, so that the resulting interactive material is not just an electronic copy of a textbook.

2. Interactivity

The word "interaction" itself is a foreign word and expresses the mutual action of two or more factors. The term "interactive" means enabling mutual communication, active input into a program or activity. Interactivity can be understood as mutual influence or a certain activity between several subjects. Interactivity is also used to refer to such types of media that allow the recipient not only to immediately respond to the received information, but also to enter its creation, control, guide the progress of work, or even complete the processed material [1].

The development of digital technologies and ICT tools has enabled and simplified communication within the internet. Depending on the applied technology, we can primarily speak of interactive communication, i.e., an immediate reaction to a stimulus, and especially two-way communication. The changes directly resulting from the integration of ICT elements into teaching consist of software applications designed to support teaching, where dynamic systems are at the forefront. These respond to changes in input conditions by changing graphics, numbers, or symbols. Today, there are already devices on the market that enhance existing applications. This level utilizes ICT and other technical means to bring deeper interactivity to the environment, such as interactive boards, moving robots, VR glasses, and so on.

3. Positive Technological Development

Creating and transforming digital worlds for student, where they can play, interact and, above all, learn, is a very current topic. The designs of virtual environments are currently motivated more by commercial needs than by a desire to positively influence the development of our students. To develop such a digital environment, it is advisable to follow some predetermined rules that will partially control our decisions. More than ten years ago, scientists at the MIT Media Lab and Tufts devised a set of recommendations called Positive Technological Development (PTD). The following selected concepts fall under PTD. PTD (Positive Technological Development) provides a framework that helps to understand how technologies can be designed and used to support positive behaviour, and how this behaviour can subsequently bring developmental assets. The theoretical model of the framework for positive technological development includes three components: individual assets, behaviour or activities mediated by technology, and applied practice. The diagram below (Fig. 1) shows how these components are interconnected and provides examples of how they can be implemented in sets.

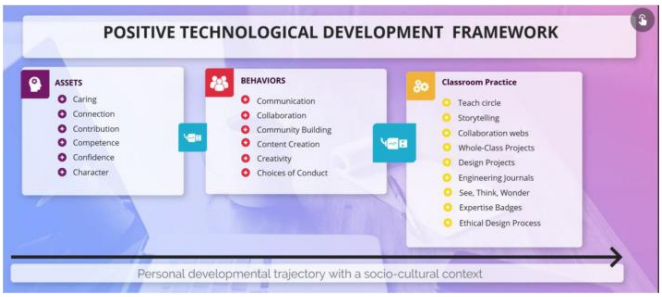

Fig. 1 *Positive Technological Development* [2]

PTD promotes collaboration instead of competition, supports shared resources and mutual care. Collaboration is integrated into the entire learning process. PTD encompasses the design of new educational technologies, technologically rich interventions, and their evaluation. Some activities could include sharing tools/materials, working on the same project, seeking help from other students, designing, and providing feedback, etc.

4. Python

Python is currently one of the most widely used programming languages, especially in the field of data analysis, and is a versatile programming language that community contributors have added an extensive library of scientific and data analysis resources to, making the range of problems that can be easily solved enormous. These resources are freely available online as numerical routine libraries, including SciPy, which includes algorithms for many classes of scientific problems such as optimization, integration, and interpolation, among others; NumPy, which includes a data structure of arrays and fast numerical routines; Matplotlib, which is a graphical package for visualizing data; and Pandas, which is built on top of Matplotlib and NumPy to allow access to both libraries with less code and fewer routines [3].

Interactive software for Python refers to a class of software that allows users to interactively run and test Python code in a convenient and user-friendly way. There are several popular interactive software tools for Python, including IDLE, Jupyter Notebook, and IPython. IDLE (Integrated Development and

Learning Environment) is a basic Python IDE that comes bundled with Python. It provides an interactive shell and a simple text editor, making it easy for beginners to write, run, and test Python code. IDLE also has features such as code highlighting, code completion, and debugging tools. Jupyter Notebook is a web-based interactive computing environment that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It supports multiple programming languages, including Python, and is widely used for data science, machine learning, and scientific research. Jupyter Notebook allows users to run code in cells and see the results immediately, making it easy to experiment with different code snippets and explore data interactively. IPython is an interactive shell for Python that provides an enhanced command-line interface with features such as syntax highlighting, tab completion, and object introspection. It also supports inline plotting and can be used as a backend for Jupyter Notebook. Overall, interactive software for Python is essential for learning, experimenting, and developing Python code in a more efficient and user-friendly way. Whether you're a beginner or an experienced developer, there's an interactive Python tool that can help you write better code and achieve your goals.

5. Result and discussion

The research shows that all interactions with visualization tools are more effective for learning than watching (non-interactive) video animations. The studies indicate that sophisticated interactivity in education has increased students' interest, but this did not translate into better academic performance.

The research sample consisted of 28 students. We obtained the responses to the questionnaire after the instruction with interactive software. Most of the participants (86%) stated that interactive instruction was better and more effective than previous teaching methods. 14% of participants described the instruction as average, and none of the participants reported that interactive instruction was worse than previous teaching methods. For the subjective measurement of cognitive load, we used a questionnaire developed by Paas in 1992, where mental effort refers to the number of cognitive capacities or resources that students allocate to meet the demands placed on them [4]. After completing the instructions, students were immediately asked to report the mental effort they invested in learning. Mental effort was rated on a 9-point Likert scale by selecting 1 (very, very low) to 9 (extremely high). It is possible to say that the use of an interactive program for teaching programming in Python had a positive impact on the cognitive load of the students. The use of interactive software reduces cognitive load, allowing students to achieve better results. Research indicates that a high level of cognitive load has a negative impact on cognitive capacity and performance [5].

6. Conclusion

Our findings that students found the interactive learning environment suitable and the interactivity itself had a positive impact on their learning are in line with the works of other authors in the same area [6,7]. For the teacher, it is necessary to master the didactics of interactive learning, which provides students with a deeper and easier insight into programming teaching. In future work, we would like to test a larger sample of students and assess the effectiveness at significant levels for knowledge transfer. In the case of simulations, we want to measure the level of impact on students and their improvement in complex programming structures. It should be noted that the sample size can affect the repeatability of the research and its conclusions. We recommend conducting a study examining the distribution of interactivity and its impact on cognitive functions.

7. References

1. I. Psenakova, *Tvorba Interaktivnych aplikacii* (TUT, Trnava, 2019)

2. M. Bers, et al. CESD 138 (2019)

3. M. Anshari, et al. EIT 22 (2017)

4. F. Paas, JEP **84**, 4 (1992)

5. J. Pimentel, et al. ESE **26**, 65 (2021)

6. B. Harrington & N. Cheng, CompScieEdu 49 (2018)

7. Y. Park, IntRevODLea **12**, 2 (2011)