

# Extension of Java with Turtle Graphics

Zoran Putnik and Marijana Grabovac

Computer graphics has entered all aspects of computer usage, especially after introduction of World Wide Web into everyday life and after the number of people who are using it increased. This emphasized the need for better quality and enlarged quantity of graphics we are faced with. At the same time, WWW induced new and powerful concept - programming language Java. This programming language, together with technical and technological improvements, brought the possibility of inclusion of larger amount of multimedia elements and network into all kinds of software.

On the other hand since the invention of programming language LOGO, turtle graphics has been recognized as an excellent educational tool. It is especially suitable for a development of procedural thinking, offering more natural way for unification of a program with data structures and serving as a ready-made prototyping tool. The language was designed at Bolt Beranek and Newman INC, in Cambridge, Massachusetts, USA, in the late 1960s [1]. The real reason for its development was testing of an idea that programming may be used as an educational discipline to teach children, especially about notions hard for understanding (mathematical concepts, experiments with abstract concepts).

In any programming language, there are two basic ways of employing computer graphics: usage of basic graphic functions and usage of turtle graphics. While the use of basic graphics functions confirms essentially to use of (any) other (kind of) functions of a given language, turtle graphics, less known and used, has its own "rules" and techniques, common for every language. Traditionally and informally, turtle graphics may be described as a process of picture creation through a movement of "an animal" (i.e. a turtle), which crawls across a screen, towing a pencil and leaving a trail [1].

Turtle graphics has been explored at the Institute of Mathematics in Novi Sad for some time, and it was at first introduced as an extension of functional programming language LispKit LISP developed at the Institute [2]. Later, research concentrated on educational values of turtle graphics in [3]. As it may be noticed, at the time, study focussed on combination of functional programming and turtle graphics. Yet, with the introduction and expansion of Java programming language, some problems existant in mentioned research, related with functional programming are naturally overcome.

Now, turtle graphics has been implemented in Java programming language as one Java applet. The implementation of turtle graphics in Java is important because it brings the expressiveness of LOGO to Java language. At the same time, implementation of turtle graphics in Java programming language implicitly introduced it into several other operating systems. Java applets, with easy control of mouse, buttons, "choices" and authority over the computer even for non-programmers enabled greater concentration on abilities of turtle graphics and their use in education.

The choice of Java programming language proved as an excellent one, since we got a language that is portable, interpreted, high-performance, simple and object-oriented, which are all characteristics of a modern programming language of a high quality.

In this paper we have implemented something that has already proved its educational and creative value. In this way we have presented it to the wider public in the environment that is secure and user friendly. Since the program was carried out in Java it can operate on any platform (operating system) without the need for recompilation or reimplementatation.

## References

- [1] Ross, P. *LOGO Programming for the IBM OC*, Addison Wesley, Wokingham.
- [2] Putnik, Z., Budimac, Z., and Ivanovic, M. *Turtle Walk Through Functional Language*, SIG-PLAN Notices, Vol. 26, no. 2, pp 75-82.
- [3] Putnik, Z., and Budimac, Z., *Bringing LOGO to the World of Purely Functional Programming*, In Proc. of 'EUROLOGO '97' - Conference, Budapest, Hungary, pp. 406-409.