



## A NEW GRAPHICAL LOGO DESIGN: LOGOTURK

**Erol Karakirik**

**Soner Durmus**

**ABSTRACT.** The development of abstract mathematical thinking is an essential part of mathematics and the geometry is regarded as a suitable domain to serve this purpose. As different technologies such as computers and graphing calculators are widely being used, curriculum developers on geometry should take these technologies into consideration. Several Logo-based computer environments have been designed to develop conceptual understanding and abstract thinking in geometry. A new graphical logo environment, LogoTurk, have been designed to eliminate some deficiencies in these environments and to provide a graphical environment in which students could explore geometric concepts and relations in different ways. The purpose of this paper is i) to present the pedagogical needs to develop a new graphical logo design, ii) to introduce the graphical features of LogoTurk meeting these needs, iii) to evaluate this new design.

**KEYWORDS.** Logo, LogoTurk, Graphical design.

### INTRODUCTION

Geometry is an abstract branch of mathematics that helps students reason and understand the axiomatic structure of mathematics. Because of the nature of concepts and relations of geometry, it is an abstract subject for most of the primary school students (NCTM, 2000). It is concerned with finding the properties and the measurements of certain geometric objects. Geometric properties are those properties of the objects that remain invariant under certain transformations when the sizes and measurements of the objects change. Carpenter et al. (1980) and Flanders (1987) claim that current geometry curricula focus on lists of definitions and properties of shapes, and learning to write the proper symbolism for simple geometric concepts. Having a relational understanding means that one should be aware of knowing why and how to do certain operations. Using relational understanding for teaching geometry emphasizes concepts, such as angles, sides, triangles etc. and analyzes the spatial relationships, such as angle measure, length, area, congruency, and parallelism. This approach helps students' to develop their conceptual understanding and improve their usage of conceptual knowledge during problem solving process.

There is a commonly accepted theory, which is based on studies of Pierre and Dina van Hiele, to explain and help us on understanding of development of geometrical thinking of students. Van Hiele's theory proposes that students move through different levels of geometrical thinking (Clements & Battista 1992). Curriculum developers and teachers should consider these levels by enriching learning environment to help students progress from one level to the next level since these levels are progressive (Burger & Shaughnessy, 1986). It is claimed that current primary geometry curricula neglects and do not promote opportunities for students to use their basic intuitions and simple concepts to progress to higher levels of geometric thought. This problem becomes more apparent in high school where students are required to employ their deductive reasoning (Hoffer 1981; Shaughnessy & Burger 1985). Deficiencies on conceptual and procedural understanding of students cause problems for the later study of important ideas such as vectors, coordinates, transformations, and trigonometry (Fey et al. 1984).

### **LOGO GEOMETRIES**

Students at early van Hiele levels need to experience with concrete materials. Action is a very important component in the development of geometrical thinking (Piaget and Inhelder, 1967). Physical actions with concrete materials are crucial for students to internalize geometric notions. Technology enables students to visualize geometric concepts and relations in a more concrete sense. For instance, geometry rods, geobord, isometric papers, symmetry mirrors etc. are some examples of technologies that might help students construct geometric ideas. Geometry standards put emphasize on focusing on the development of careful reasoning and proof using definitions and established facts (NCTM, 2000).

Logo geometry environments claim to facilitate the developmental process of geometrical thinking. For example, students might transfer their actions to logo environment on the computer via giving directions to the turtle on the screen. Hence, through monitoring actions of the turtle, they might internalize their own physical actions as to develop geometrical interpretations of actions at hand.

Logo environments are designed to achieve three major goals (Clement & McMillen, 2001, pp.14-15): i) achieving higher levels of geometric thinking, ii) helping students learn major geometric concepts and skills, and iii) developing power and beliefs in mathematical problem solving and reasoning. Developers of Logo Geometries have assumed that curriculum has three strands: Paths, shapes, and motions. Relational understanding can be based on these three strands. The rationale behind developing Logo environments is to facilitate constructivist philosophy of learning which emphasize active involvements of students during teaching-learning process. The details of this issues is discussed by Karakirik and Soner (2005).

### FIRST PHASE OF LOGOTURK

The development of LogoTurk environment passed through two phases. In the first phase, a logo environment that accepts classical logo commands in its traditional form in different languages were implemented (Karakirik and Durmus, 2005). For instance, Fig. 1 shows how to write a procedure producing a hexagon of length 60 pixels with traditional Logo commands in LogoTurk.

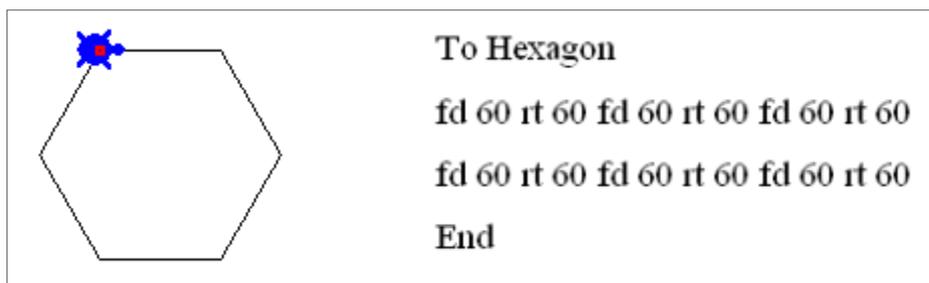


Figure 1. Drawing a hexagon in LogoTurk

One can also construct the same hexagon with the following code using a loop.

```
Repeat 6 [ forward 60 right 60]
```

Similarly, Table 1 shows how to construct any regular polygon with a loop.

Equilateral Triangle	Square	Regular n-gon
To Triangle	To Square	To NGon(_n)
Repeat 3 [ forward 50 right 120]	Repeat 4 [ forward 50 right 90]	Repeat _n [ forward 50 right 360/_n]
End	End	End

Table 1. Procedures of Creating Regular Polygons

LogoTurk adopts a different way of defining a procedure with the help of a procedure editor. This design also enabled testing of each procedure separately and minimized the students' loss of data. One can add, delete, rename, run and stop each procedure separately by related menu items and shortcuts in LogoTurk. In addition, it has an error detection mechanism which enables both detection and removal of small typographical errors. Karakirik and Durmus (2005) provide the details of the implementation of the first phase. LogoTurk enables students and teachers to pose and solve their own problems. One can construct, for instance, creative figures with the help of iterations (See Appendix A) or a desired specific shape such a house (See Appendix B). It also allows students to save configurations and their sequences of actions.

The modifications made in the first phase did not remove some deficiencies encountered in the classical Logo environment. We have translated the classical Logo commands to Turkish

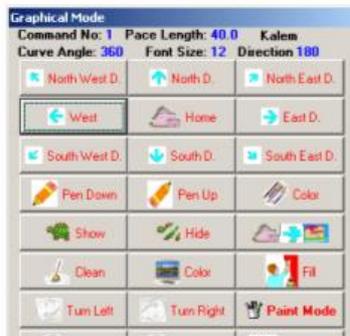
in order to remove the language barrier since Logo strictly relies on syntax to carry out simple commands. Although we were able to integrate all classical Logo commands in our environment, a need arose to develop a totally new approach to eliminate the difficulties with syntax and the traditional implementation of Logo itself. Hence, we developed a new graphical interface of LogoTurk in the second phase.

### PEDAGOGICAL CONCERNS ABOUT LOGOTURK

This section provides the pedagogical needs to develop new graphical design for classical Logo Environments. We will outline the pedagogical issues by outlining the new features of the graphical version of the LogoTurk.

There is always a concern that integrating graphical elements to any software packages could reduce users' cognitive involvement with the task at hand and may distract their attention from intended objectives. Some also may regard the reducing Logo commands to graphical elements as educationally unfavorable because of the aforementioned concerns. The graphical elements of LogoTurk is designed in a way that students need to utilize their conceptual knowledge. For instance, mouse movements that enable easy modification of drawn figures by dragging certain points are disabled. Instead, students are required to use classical Logo commands to make the necessary changes.

The graphical version of LogoTurk dynamically links different representations and maintain a tight connection between pictured objects and symbols. A student can act both as a turtle moving on the screen and as a person monitoring out of the screen by using different modes giving different meanings to directions. For instance, if a student wants to move the turtle to north, he/she can use either "North D." button as a person monitoring out of the screen (See Fig. 2) or "Left" button as a turtle moving on the screen (See Fig. 3 ). The switch between these two modes may enable students to reflect on their actions and may change their perceptions about the relative meanings of angles and directions. Therefore, LogoTurk could be used to assess students' conceptual knowledge about the relative meanings of angles and directions.



**Figure 2.** As a person monitoring out of the screen : Using real directions mode



**Figure 3.** As a turtle moving on the screen: Using relative directions mode

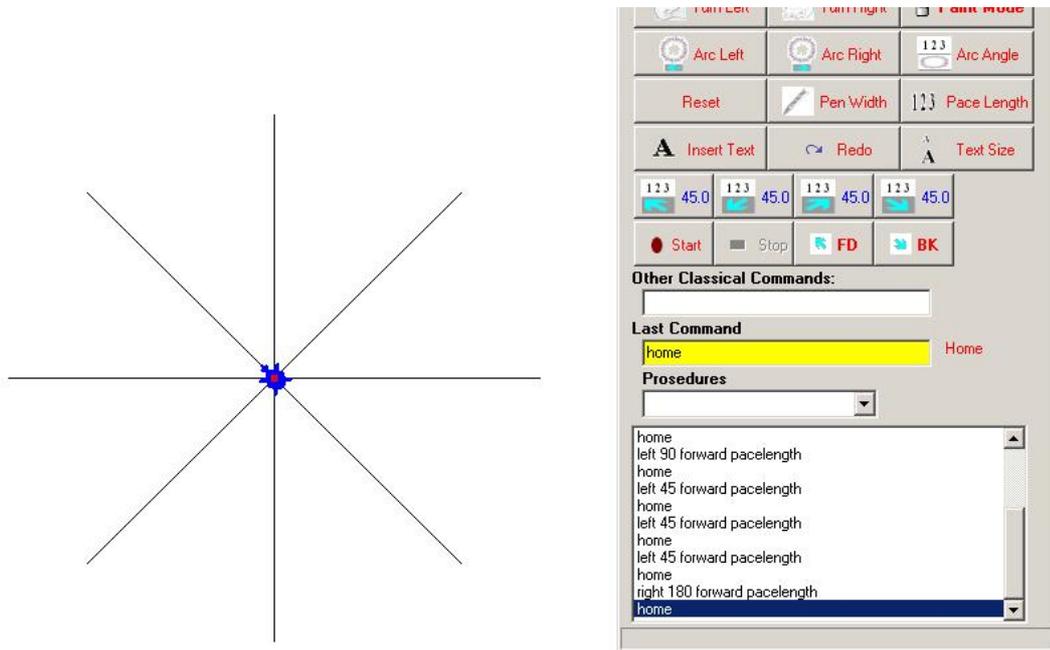
LogoTurk as a computer manipulative might provide an environment including tasks that cause students to see conflicts or gaps in their thinking. For instance, requiring students to draw a simple house with two windows, a door and a roof in aforementioned two modes could be a real eye-opener.

Although constructing a house as a person monitoring out of the screen resembles to drawing a house picture in "Paint" program, constructing a house as a turtle moving on the screen requires using one's conceptual knowledge about the relative meanings of angles and directions and seemingly much more difficult.

The biggest advantage of the graphical version of LogoTurk is eliminating students' dependency on both syntax and semantics of the classical Logo environments. One can get rid of the syntax of Logo by pressing certain buttons instead of writing a code to give a command to the turtle. Similarly, one can avoid the semantics of Logo by switching between relative and real mode of directions. Hence, LogoTurk as a mathematical tool allows students to develop increasing control of their actions. Although the classical Logo is designed for emphasizing the relative meaning of the directions requiring one to see himself/herself as a turtle moving on the screen, it might be beneficial to make switches in the real world. For instance, an architect might employ both modes to construct certain parts of his/her design.

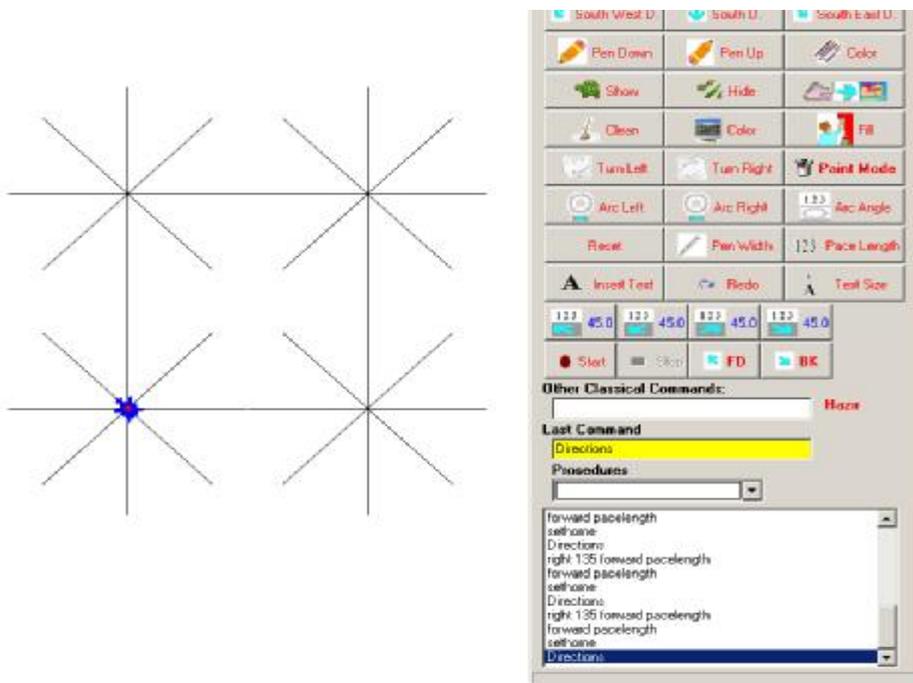
The graphical version of LogoTurk also supports creating procedures by dynamically grouping a number of actions under a macro name. One can start a macro definition by pressing the "Start" (Macro) button and stop it by pressing the "Stop" (Button). The macro is defined relatively with respect to the mode (See Fig. 7). Students could re-use the created macro either by a name or making a selection from a list of defined macros. It is claimed that students could better appreciate the meaning of a procedure in this way as a group of repeated actions without having any difficulty with the syntax of writing a procedure. Furthermore, LogoTurk also supports writing procedures in a separate text window in case complex figures need to be constructed with the help of iterated commands.

The graphical version of LogoTurk helps to visualize the effects of the classical logo commands. Every action performed by pressing a button is recorded and translated to the classical logo commands. The history of the action were also displayed at the bottom(See Fig 7). Therefore, It is claimed that students could grasp easily the meanings of the classical commands while they construct their geometric figures with graphical elements. Some graphical elements, such as "Arc left" and "Arc Right", produce actions that could be performed by a set of classical logo commands. LogoTurk also introduced a completely new command "Sethome" to set any point on the screen as a reference point for further operations. For instance, Fig. 4 shows how to construct a directions macro showing the usual 8 directions employing the classical "home" command. However, calling this macro always produces the same figure since the reference point for home is predefined.



**Figure 4.** The macro definition of "Directions" with "Home" command in LogoTurk

However, Fig. 5 presents a complex figure produced by calling the defined "Directions" macro at different reference points (i.e. home) set by "SetHome" command. This gives flexibility to recall a figure at any part of the design.



**Figure 5.** The usage of newly introduced "SetHome" command in LogoTurk

LogoTurk also provides a facility to undo the last action to improve user-friendliness. However, the provision of such a facility might contradict the initial Logo philosophy which aims students to find the correct action by trial and error method. Hence, the undo action is hidden in the graphical interface, could be executed by the classical Windows shortcut (Ctrl-Z) and could be activated and deactivated by related menu items. The rationale for adding this feature was to encourage students to try different geometric figures without worrying about the mistakes that he/she can not recover. This also resembles the similar facilities found in "Paint" programs so that students may feel comfortable with using the system. It might help creating implicit associations between geometry and other drawing programs that some professionals such as architects and engineers use.

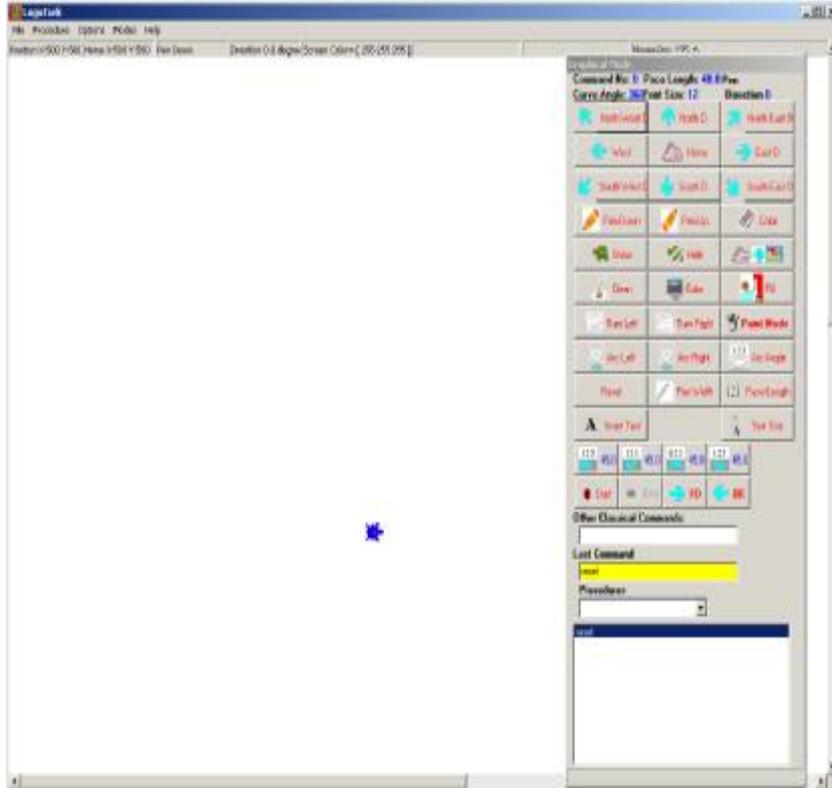
### **THE DESIGN OF GRAPHICAL VERSION OF LOGOTURK**

Design is a very important part of the development of any system. The best system could be described as a transparent system that does not allow the medium used, the computers in this context, interfere with the task but enhance the user's experience without changing the nature of the task. A crucial aspect of the design of a system is the continuous and iterative nature of the design process that is carried out with the help of experimental studies. However, there are some design issues that can not easily be resolved by experimental studies. There might be several possible working versions of the same system. Hence, priorities and the specifications of the system should be determined in advance by considering the requirements and the convenience of the task at hand as far as possible.

Prior conventions and the author's own preferences might be used to make some design issues. However, employment of evaluators that are expert on the application domain is also necessary to detect some problems and to decide some issues with the system. Three to five evaluators are considered as optimal since different evaluators could find different problems (Nielsen, 1994). Guidelines for the user interface and some checklists might also be beneficial in the early stages of the development. For instance, Nielsen (1994) puts forward such a 10 items checklist, called heuristic evaluation for this purpose. It propagates a minimalistic and simplistic design based on functionality of the system and also includes items for checking consistency, flexibility, documentation, diagnosing and recovering errors, and visibility of system status and utilization of visual clues where appropriate. These considerations were taken into account during the development of LogoTurk. This section gives the details of the new design of LogoTurk.

The classical Logo commands were integrated into the design of the graphical version of LogoTurk by the provision of an area named "Other Classical Commands" to enter them manually (See Fig. 6). This might give students the flexibility to use their accustomed way of

using Logo so that they may feel comfortable. Furthermore, students may choose to switch to the classical mode completely by a related menu item. Graphical elements in the form of icons were developed for most commonly used Logo commands (See Appendix C). It is possible to switch between graphical and classical mode by choosing related menu items. There is also an option to simulate turtle's actions in a quick or slow fashion. This option abolishes to enter classical "wait" command to see the effects of each individual turtle action.



**Figure 6.** Graphical Version of LogoTurk

These icons, when pressed, converts users' actions into classical Logo commands and pass them to Logo engine be executed. One icon press may produce a block of Logo commands. For instance, pressing "Arc Left" icon produces the following block of Logo commands:



**repeat 36 [ lt 10 fd 40 ]**

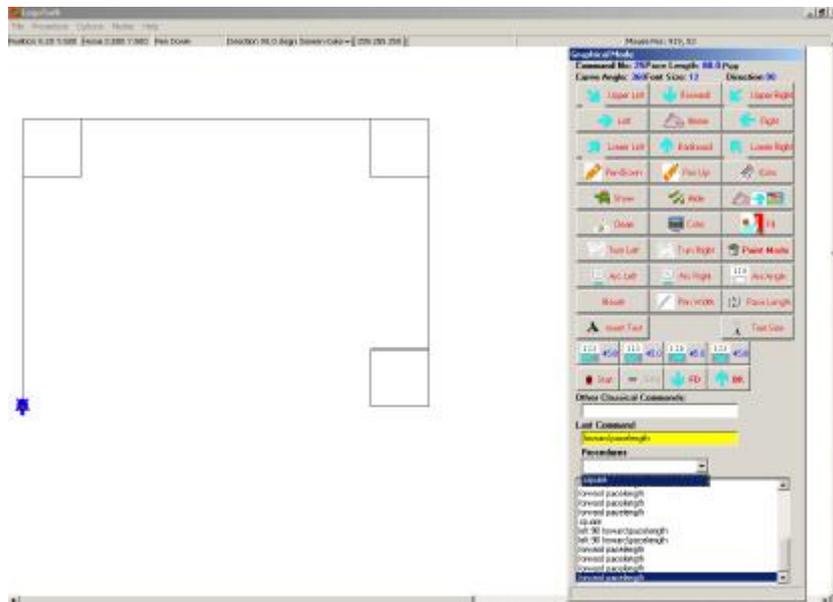
There are also some global variables to direct turtle's actions in a uniform way. For instance, the pace length variable defines the length of turtle's pace for moving any direction. Moving the turtle forward one pace results in moving the turtle forwards as the value of the pace length variable.

The history of turtle's precious actions could be seen in a combo-box at the lower part of the graphical interface. This combo-box and the "last command" section of the interface was

updated after new Logo commands were produced as a result of users' key presses. There is also a command line to enter any classical Logo commands without using the graphical elements. This facility is useful especially for creating repetitive actions of turtle. For instance, the following command draws a regular hexagon:

**repeat 6 [ Lt 60 fd 50 ]**

Procedures are defined in classical Logo in order to reproduce a set of commands resulting in a certain figure or group of actions. Likewise, procedures are simulated as macros in LogoTurk. A number of actions can be recorded as a macro by defining a starting and ending points by pressing related icons. For example, pressing four times Left button produces a square of one pancelength. One can record it as a macro by giving it a name as a "square" and several "squares" can be constructed by recalling this macro from procedures section shown in Figure 7.



**Figure 7.** A macro definition of a square in LogoTurk

There are two different set of icons for 8 directions that controls the motions of the turtle. One set of icons are static denoting the known directions, namely east, north east, north, north west, west, south west, south and south east , east, north east, north, north west, west, south west, south and south east. The second icon set controls the motions of the turtle as if the user moves as a turtle itself and the directions that icons denote change with respect to the turtle's current position and direction. The icons in the first set always show the same direction while the directions of icons in the second change after every movement of the turtle. Different names are used for icons to denote the differences between two sets. While the first is set claimed to be proper for easy construction of certain shapes regardless of turtle's position such as a house, the second set is much more consistent with Logo paradigms and proper for drawing and seeing

geometric shapes such as square. For instance, Figure 8 shows two different ways of constructing a square with two different icon sets. The absolute direction with "setheading" command is used in left part of Figure 7 while the relative direction with "left" command is used in the right part of the figure. There is a need to investigate the effects of these two different modes on students' conceptual understanding of geometric relations existing in the figures.

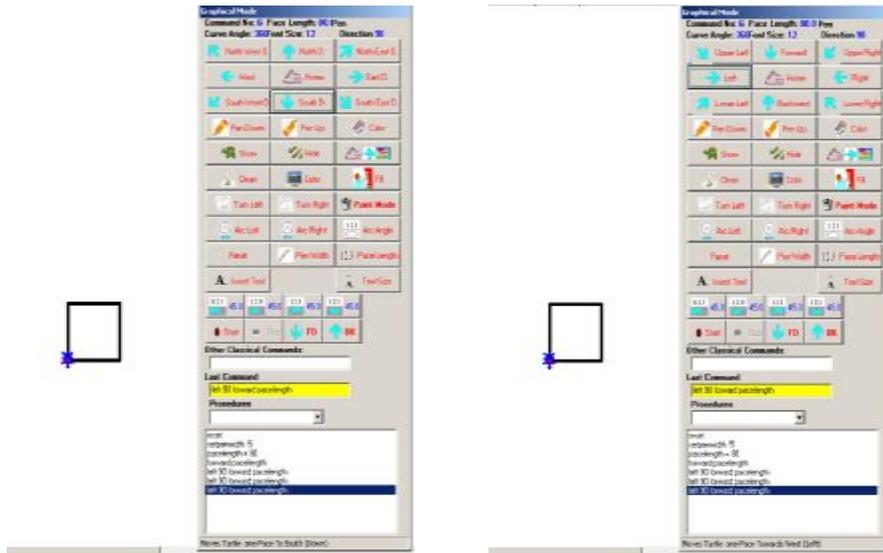


Figure 8. Drawing a square using two different set of icons in LogoTurk

## AN EVALUATION OF LOGOTURK

We successfully designed a new graphical version of Logo Environment. Our design differs from other graphical designs in a way that it not only includes the all classical Logo commands with graphical elements but also introduces new ways of experiencing geometry in Logo environment.

Our new design provides an environment where a student can act both as a turtle moving on the screen and as a person monitoring out of the screen. This might help students grasp different interpretations of relative meanings of angles and directions. It is suggested that students might create geometric constructs resembling what architects and engineers make. So, geometry might be seen as a part of the real life. Our new design has the potential to enhance students' geometry experiences and enrich their geometrical thinking. So, it is claimed that designing user friendly interfaces for Logo may change students' perception of Logo and made them focus on more conceptual oriented geometrical tasks. New studies should be designed to examine the effect of newly added graphical elements of Logo on conceptual understanding.

Our new graphical design might affect the curriculum of the elementary geometry and the way geometry is being perceived and taught. Curriculum developers and instructional designers might take our pedagogical concerns into the consideration so that they can benefit from the advantages of our design. Further experimental studies need to be designed to assert our claim.

#### REFERENCES

- Burger, W. F., & Shaughnessy, J. M. (1986). Characterizing the van Hiele levels of development in geometry. *Journal for Research in Mathematics Education*, 17: 31-48.
- Clements, D. H., & McMillen, S. (2001). *Logo and Geometry*. Journal for Research in Mathematics Education Monograph Series, Arlington, VA: National Science Foundation.
- Clements, Douglas H., & Battista, Michael T. (1992). *Geometry and Spatial Reasoning*. In Douglas A. Grouws (ed.), *Handbook of Research on Mathematics Teaching and Learning*, 420-64. New York: Macmillan.
- Carpenter, Thomas P., Mary, K., Corbitt, Henry S., Kepner, Mary M. Lindquist & Robert, E. Reys. (1980). *National Assessment*. In Elizabeth Fennema (ed.), *Mathematics education Research: Implications for the 80s*, Alexandria, Va.: Association for Supervision and Curriculum Development.
- Fey, James, Atchison, William F., Richard, A. Good, Heid, M. Kathleen, Johnson, Jerry; et al. (1984). *Computing and Mathematics: The Impact on Secondary School Curricula*. College Park, Md.: University of Maryland.
- Flanders, James R. (1987). How Much of the Content in Mathematics Textbooks is New. *Arithmetic Teacher* 35: 18-23.
- Feurzeig, Wallace & Lucas, George. (1972). *Logo--A Programming Language for Teaching Mathematics*. *Educational Technology* 12: 39-46.
- Hoffer, Alan. (1981). *Geometry Is More than Proof*. *Mathematics Teacher* 74: 11-18.
- Howe, J. A. M., O'Shea, T., & Plane, F. (1980). *Teaching mathematics through Logo programming: An evaluation study*. In R. Lewis & E. D. Tagg (Eds.), *Computer assisted learning: Scope, progress and limits*, pp. 85-102. Amsterdam NY: North-Holland.
- Karakirik, E. & Durmus, S. (2005). *An Alternative Approach To Logo-Based Geometry*. *The Turkish Online Journal of Educational Technology - TOJET*, 4 (1):1-14.
- Kieran, C., & Hillel, J. (1990). *It's tough when you have to make the triangles: Insights from a computer-based geometry environment*. *Journal of Mathematical Behavior*, 9: 99-127. .
- National Council of Teachers of Mathematics. (2000). *Principles and Standards for School Mathematics*, Reston: The National Council of Teachers of Mathematics, Inc.
- Nielsen, J. (1994). *Heuristic evaluation*. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*

(pp 25-64). New York, NY: John Wiley & Sons.

Shaughnessy, J. Michael & William, F. Burger. (1985): Spadework Prior to Deduction in Geometry. *Mathematics Teacher* 78: 419-428.

Piaget, Jean & Inhelder, Barbel. (1967). *The Child's Conception of Space*. New York: W. W. Norton & Co.

**Karakirik, Erol**

Faculty of Education

Abant Izzet Baysal University

Bolu / Turkey.

Email: [karakirik@hotmail.com](mailto:karakirik@hotmail.com)

**Durmus, Soner**

Faculty of Education

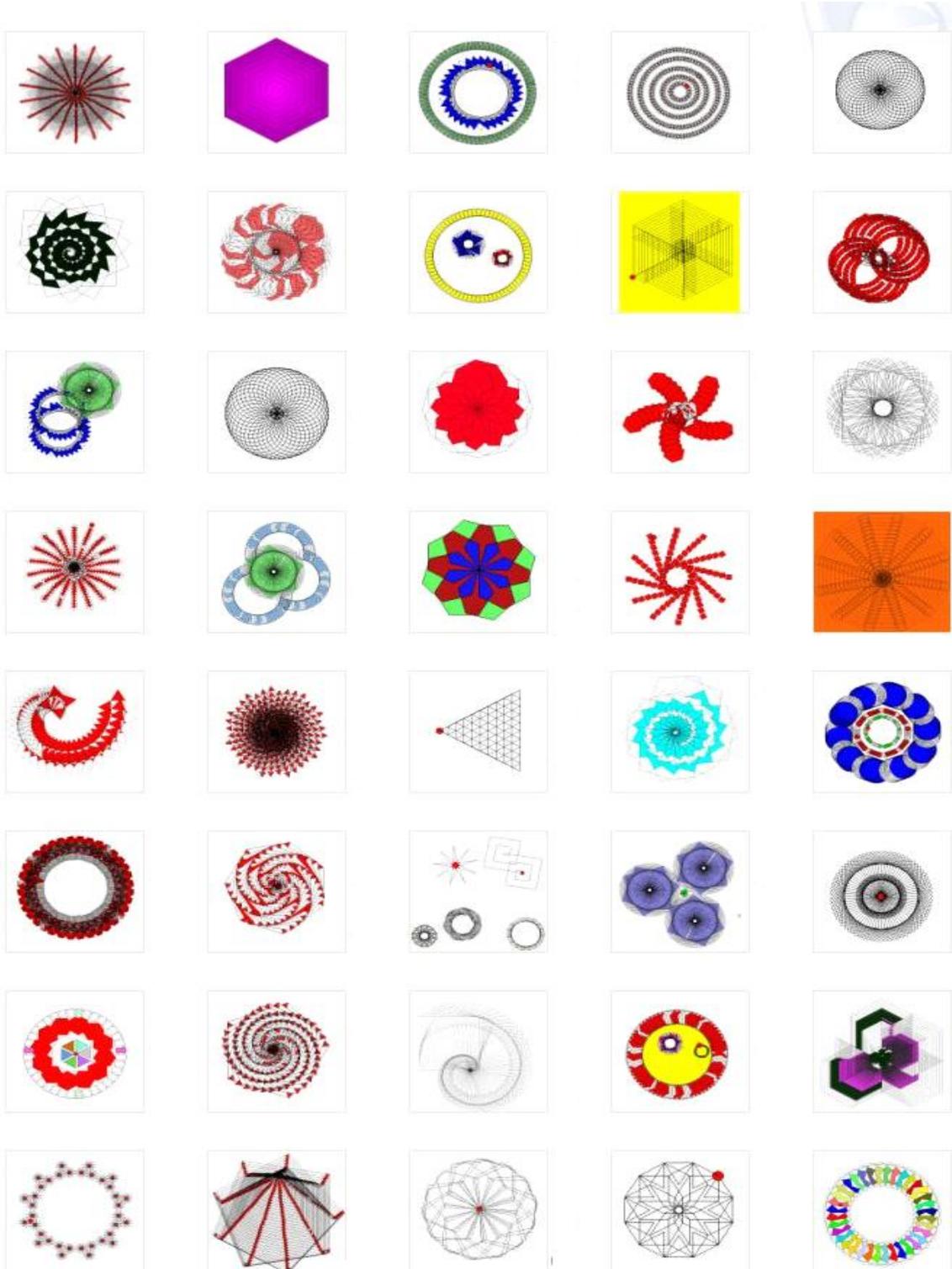
Abant Izzet Baysal University

Bolu / Turkey.

Email: [sdurmus@ibu.edu.tr](mailto:sdurmus@ibu.edu.tr)

APPENDIX A

Some figures created by using LogoTurk.



**APPENDIX B**

Some house figures created by using LogoTurk.



## APPENDIX C

## Graphical Elements of LogoTurk.

Icons	Logo Comands	Explanation
	Forward ?	Moves Turtle one pace forward
	Back ?	Moves Turtle one pace backward
	-	Moves Turtle one pace to some directions namely northwest, north, north east, south west, south, south east, west and east
	Setfillcolor [ ? ? ? ] Fill	Fills a closed region with the selected color
	PenDown	Sets the pen mode Down
	PenUp	Sets the pen mode Up
	SetPenWidth ?	Sets the width of the pen to the selected size
	SetPenColor [ ? ? ? ]	Sets the color of the pen to the selected color
	ClearScreen	Cleans the Screen
	HideTurtle	Hides the Turtle
	ShowTurtle	Shows the Turtle
	Left ?	Turns the Turtle to the Left with ? degrees
	Right ?	Turns the Turtle to the Right with ? degrees
	SetScreenColor [ ? ? ? ]	Sets the color of the screen to the selected color
	Label ?	Inserts text at the current location of the turtle
	SetFontsize ?	Sets the size of the font to the selected size
	Home	Sets the Turtle position to home
	-	Sets the current position as the new home for the turtle
	-	A block of commands to produce an arc to the right with a specified arc length
	-	A block of commands to produce an arc to the left with a specified arc length
	-	Specifies the arc length for drawing arcs to the left or to the right
	-	Specifies the length of the pace of the turtle for moving any direction
	-	Adjusts the angle to return for certain directions namely northwest, southwest, northeast and southeast
	-	Starts a macro to define a procedure
	-	Stops the macro and asks for a name for the procedure
	Penpaint	Sets the paint mode to Penpaint
	Penerase	Sets the paint mode to Penerase
	-	Starts the macro definition
	-	Stops the macro definition