

AN ALTERNATIVE APPROACH TO LOGO-BASED GEOMETRY

Erol KARAKIRIK, Soner DURMUS
Abant İzzet Baysal Üniversitesi, Eğitim Fakültesi
durmus_s1@ibu.edu.tr, karakirik@hotmail.com

ABSTRACT

Geometry is an important branch of mathematics. Geometry curriculum can be enriched by using different Technologies such as graphing calculators and computers. Logo-based different software packages aim to improve conceptual understanding in geometry. The goals of this paper are i) to present theoretical foundations of any compute software developed to enrich primary and secondary school geometry curriculum, ii) to introduce main features of Logo, iii) to introduce a new version of Logo called LogoTurk, and iv) to present some recommendations for future research.

Keywords: Geometry, Logo, LogoTurk

INTRODUCTION

Geometry is an important branch of mathematics. Geometry is seen as the place in the school curriculum where students learn to reason and understand the axiomatic structure of mathematics (NCTM, 2000). Since current elementary school geometry curriculums focus on lists of definitions and properties of shapes, and learning to write the proper symbolism for simple geometric concepts (Carpenter et al., 1980; Flanders, 1987), it underemphasizes relational understanding that means knowing both what to do and why. For example, learning lists of properties of quadrilaterals and triangles is as important as being involved in the process of developing and using a property-based conceptual approach for reasoning about these shapes (Battista, 2002). This approach uses such concepts as angles, sides, angle measure, length, area, congruency, and parallelism to describe and analyze spatial relationships in and among these shapes. In particular, when we define an equilateral triangle as being three-sided figure that has three equivalent angles and all sides the same length, we are using the concepts of sides, length, and angle measure to describe the spatial relationships that differentiate equilateral triangle from other triangles.

Current educational theories emphasize active involvement of students on teaching-learning process. Through abstraction and reflection during this process, students construct their mathematical knowledge. Mathematics-geometry instruction should facilitate learning environment that support inquiry, problem solving and sense making in which students invent, test, and refine ideas to build complex, abstract, and powerful mathematical meanings (Battista, 2002). There are available computer software packages aiming to enrich learning environments to achieve these goals. This paper aims i) to present theoretical foundations of any computer software developed to enrich primary and secondary school geometry curriculum, ii) to introduce main features of Logo, iii) to introduce a new version of Logo called LogoTurk, and iv) to present some recommendations for future research.

GEOMETRY

Geometry as branch of mathematics that has special characteristics, those need to be taken into consideration. Objects and properties of objects in geometry are extremely abstract for primary school students. Students are expected to recognize shapes and comprehend their properties. Students' developmental stage is the key that one should be aware of. There are theories developed based on studies of Piaget's and Pierre and Dina Van Hiele's to explain and help us on understanding of development of geometrical thinking. In Piaget's work, there are two major themes related to geometrical thinking. Firstly, development of geometric ideas follows a definite order which is more logical than historical. Topological relations (connectedness, continuity) develop first, followed by Euclidean (angularity, parallelism, area) relations. It develops over time by integrating and synthesizing these relations to their existing schemas. Secondly, mental representation of space develops through progressive organization of the student's motor and internalized actions. Ideas about space evolve as students interact with their environments. These two themes are supported by research (Clements & Battista, 1992).

Van Hiele Theory proposes that students move through different levels of geometrical thinking. These levels are as follows:

- ✓ Level 0 (Prerecognition): Since students do not comprehend visual characteristics of shapes, they are unable to identify many common shapes.
- ✓ Level 1 (Visual): They can only recognize shapes as whole images. A given figure is square, for example, because it looks like a window. Students do not think of properties of shapes.

- ✓ Level 2 (Descriptive/Analytic): Students by observing, measuring, drawing, and model making can recognize and characterize shapes by their properties. For instance, a square can be thought as a figure that has four equal sides and four right angles. Students can not form classes of figures at this level. For instance, a student may not see a square as a rectangle.
- ✓ Level 3 (Abstract/Relational): Students can distinguish a shape based on certain properties which it has. They can provide logical arguments, for example, the sum of the angle measures of a quadrilateral is 360, because it consists of two triangles. As students discover such properties they go beyond definitions of certain figures, for example, a square can also be seen as a rectangle because a square satisfies all conditions for being a rectangle.
- ✓ Level 4 (Axiomatic): Students can establish theorems with an axiomatic structure. They recognize undefined terms, definitions, axioms, and theorems. They can produce certain number of statements to prove a statement.

According to Van Hiele Theory, geometric thinking levels of students in elementary and middle school are at most level 3. Thinking at level 4 is necessary for high school geometry. According to the Van Hiele theory, the levels are progressive that students move from one level of thinking to the next. Curriculum developers and teachers should take these levels into consideration by enriching learning environment to help students to progress to a next level (Burger & Shaughnessy, 1986).

There are numerous assessment reports revealing that students fail to learn basic geometric concepts especially geometric problem solving (Kouba et al., 1988; Stigler, Lee & Stevensen, 1990; the International Study Center, 1999). The current elementary and middle school geometry curriculum do facilitate opportunities for students to use their basic intuitions and simple concepts to progress to higher levels of geometric thought. Students going through such experience in elementary school do not have the necessary geometric intuition and background for a formal deductive geometry course in high schools (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).

GEOMETRY AND TECHNOLOGY

NCTM's (2000) geometry standards put emphasize on focusing on the development of careful reasoning and proof using definitions and established facts. Geometric shapes and their properties can be visualized by manipulatives. Students at level 0, 1 and 2 need to experience with concrete materials. Concrete materials such as geometry rods, geobord, isometric papers, symmetry mirrors etc. are supposed to help students construct geometric ideas. Using manipulatives benefits students across grade level, ability level, and topics which using manipulative makes sense for that topic (Driscoll, 1983; Sowell, 1989; Suydam, 1986). Clements and McMillen (1996) proposed that using Manipulatives does not always guarantee conceptual understanding: In one study, students not using manipulatives outperformed students using manipulatives on a test of transfer (Fennema, 1972). Students sometimes used manipulatives in a rote manner (Hiebert & Wearne, 1992). Clements and McMillen (1996) claims that student often fail to link their action with Manipulatives to describe the actions. Some computer manipulatives may be more beneficial than any physical manipulative. Any program having the following features can be thought as beneficial computer manipulative (Clements & McMillen, 2001, p.76). They

- ✓ have uncomplicated changing, repeating, and undoing actions,
- ✓ allow students to save configurations and sequences of actions,
- ✓ dynamically link different representations and maintain a tight connection between pictured objects and symbols,
- ✓ allow students and teachers to pose and solve their own problems, and
- ✓ allow students to develop increasing control of a flexible, extensible, mathematical tool. Such programs also serve many purposes and help form connections between mathematical ideas.

Selecting and using proper computer manipulative in learning environment should consider the following recommendations (Clements and McMillen, 2001, p.77):

- ✓ Use computer manipulatives for assessment as mirrors of students' thinking,
- ✓ Guide students to alter and reflect on their actions, always predicting and explaining,
- ✓ Create tasks that cause students to see conflicts or gaps in their thinking,
- ✓ Have students work cooperatively in pairs,
- ✓ If possible, use one computer and a large-screen display to focus and extend follow-up discussions with the class,
- ✓ Recognize that much information may have to be introduced before moving to work on computers, including the purpose of the software, ways to operate the hardware and software, mathematics content and problem solving strategies, and so on,

- ✓ Use extensible programs for long periods across topics when possible.

LOGO GEOMETRY

Elementary school geometry should focus on the study of objects, motions, and relationships in a spatial environment (Clements & Battista, 1986; Trafton & LeBlanc, 1973). First experiences with geometry should emphasize informal study of physical shapes and their properties and have as their primary goal the development of students' intuition and knowledge about their spatial environment. Subsequent experiences should involve analyzing and abstracting geometric concepts and relationships in increasingly formal settings.

Because current elementary geometry curricula focus only on identification of figures and the use of geometric terms (Kouba et al., 1988), little opportunity arises for geometric problem solving. Students have little chance to develop their spatial thinking, a skill that should have primary importance in the geometry curriculum. Students encounter little opportunity to analyze and reconceptualize substantive geometric ideas. Another deficiency in the current curriculum is that it does not always emphasize conceptualizations of topics that are most useful in the later learning of mathematics. For instance, the concept of angle normally encountered in elementary school is that of a union of two rays with a common endpoint, the same formal definition used in high school geometry. However, in trigonometry and calculus, an angle is thought of as a rotation. Existing elementary school geometry curricula do not address this second aspect of the angle concept, even though the latter aspect seems more closely related to navigation, "one of the most widespread representations of the idea of angle in the lives of contemporary Americans" (Papert, 1980, p.68).

Logo geometry is developed to achieve three major goals (Clement & Battista, 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 Geometry have assumed that curriculum has three strands: Paths, shapes, and motions. Relational understanding can be based on these three strands.

Students' movements are recorded as paths and Logo has special commands to simulate the same movements. For instance, a command such as "FD 40" draws a straight path 40 units long. Logo has also commands that inverse this process. For instance, a command such as "BK 40" draws a straight path 40 units long in the opposite direction of previous direction which is a sort of undoing of the command "FD 40". Logo commands help students learn to connect the idea of undoing a sequence of turtle commands to the related idea of undoing a sequence of arithmetic operations. For example, questions like "A number multiplied by 3, then added 4, if the result is 19, what should be this number?" can be solved by reversing operations addition and multiplication.

Students can write different procedures to move the turtle to draw closed or nonclosed, straight and nonstraight, simple and nonsimple paths. As paths are drawn, need for turning, which is a basis for understanding angle, may arise. Students develop conceptual understanding by relating turn and angles. Students apply series of commands to draw a path. During this process, they i) learn to reflect on the Logo commands and correspondence path these command produce, ii) also correct commands related to these paths, iii) correct the procedures, iv) gain an insight of nature their errors and the ways they correct them, and vi) learn to apply this problem solving approach to different context.

Second strand of Logo geometry is shapes. Shapes can be constructed by using simple paths. Logo helps students see shapes as combinations of paths. Students start to analyze shapes in terms of their components and properties. First, students identify shapes in their environment and describe and classify regions, including faces of solids. Second, students discuss properties of shapes. Finally, Students plan and write several Logo procedures to draw shapes that they observed in their environment. As Van Hiele levels are taken into consideration to help students construct the shapes by using their attributes, they can construct squares, rectangles, equilateral triangles, and regular polygons in an order by Logo procedures.

Constructing regular polygons requires knowing the angle concept. Angle can be thought as a path created by a forward move, a turn, and another move. As angle concept is formed, students need to consider amount of turn which is called an angle measure. Number of sides and measure of interior angles of polygons are key attributes to distinguish regular polygons. Students need to develop conceptual understanding of these attributes before writing procedures on Logo. Differences among regular polygons once reduced to number of sides and angle measures, it may not be difficult to write Logo commands. For example, square, hexagon, and octagon can be drawn by following Logo procedures:

Square

To square
Repeat 4 [forward 40 right 90]
End.

Octagon

To octagon
Repeat 6 [forward 40 right 60]
End.

Hexagon

To hexagon
Repeat 8 [forward 40 right 45]
End.

In the above example, number of repetitions and the measure of the angle used to make a turn after each forward movement determines the geometric shape constructed. For instance, 4 repetitions and 90 degree turns construct a square while 8 repetitions and 45 degree turns construct a hexagon. The total degree of the turn made in each repeat block is equal to 360 degrees. This, by itself, denotes that the constructed figure is a regular polygon.

Last strand of Logo geometry is motions. The main idea of motion (transformational) geometry is that there are an infinite number of figures congruent to a given figure and that these figures may be related by a combination of geometric motions. Fundamental motions are slide, turns, and flips. The concept of congruence is based on these three motions. Any combination of these motions applied to any figure preserve the main attributes of this figure. Prior to using these concepts on Logo, students need to know that two figures are concurrent if they have the same size and shape (i.e., if and only if one fits on top of the other exactly). Two congruent figures are constructed if, and only if, there is a sequence of slides, flips, or turns that moves one onto the other. Logo commands are available to help student develop these related motion concepts. Logo "provides an operational universe within which students can define a mathematical process and then see its effects unfold. It is accessible to very young children for simple tasks, yet its operations can be systematically extended to express problems of considerable complexity" (Feurzeig & Lukas, 1972, p.39).

To draw a figure in Logo, students devise a set of movement instructions for the turtle. They must determine angle measures and lengths of line segments. They can be asked to analyze the figure and break it into smaller parts that are more easily constructed. Thus, they are constantly involved in geometric problem solving. Such Logo activities encourage students to identify goals and strategies before making overt moves toward a problem solution, create efficient problem representations, make executive decisions, and debug algorithms all of which are problem-solving skills too seldom explicitly taught in the schools.

In the words of Papert, "The computer allows, or obliges, the child to externalize intuitive expectations. When the intuition is translated into a program it becomes more obtrusive and more accessible to reflection" and can thus be used as material "for the work of remodeling intuitive knowledge" (1980, 145). Therefore, in the context of Logo, the teacher can help students elaborate their intuitions about the concept of rectangle by focusing their attention on its properties and by embellishing those intuitions with verbal labels and descriptions. Such elaboration is essential for progressing toward level 2, the descriptive-analytic level, in the Van Hiele hierarchy. Moreover, by designing a rectangle procedure with inputs, students begin to build intuitive knowledge about the concept of defining a rectangle. Through such a sequence of Logo-based experiences, not only are students progressing into higher levels of geometric thinking in the Van Hiele hierarchy, but also they are building conceptual structures about rectangles that can be useful in other situations, such as drawing quadrilaterals, triangles, or regular polygons. They are thus learning geometry relationally.

CONCRETE TO ABSTRACT

According to Piaget & Inhelder (1967), action is of paramount importance in the development of geometric conceptualizations. The child "can only 'abstract' . . . the idea of a straight line from the action of following by hand or eye without changing direction, and the idea of an angle from two intersecting movements" (p.43). Indeed, physical actions on concrete objects are necessary. But students must internalize such physical actions and abstract the corresponding geometric notions. Logo can facilitate this process, thus promoting a transition from concrete experiences with geometric ideas to abstract reasoning. For example, by first having children form paths by walking, then using Logo, children can learn to think of the turtle's actions as ones that they themselves could perform. They seem to project themselves into the place of the turtle. In so doing, they are performing a mental action--an internalized version of their own physical movements.

Because children understand beginning spatial notions in terms of action and because the mathematical concept of path can be thought of as a record of movement, it seems natural to emphasize this concept in the beginning study of geometry. For example, having students visually scan the side of a wall, run their hands along the edge of a rectangular table, or walk a straight path will help them develop an intuitive concept of straightness. But in Logo, the essence of this abstract concept can be brought to a more explicit level of awareness as a "turtle path that has no turning." Because the concept is explicit and reformulated in a more formal and precise language, it can be internalized in a more abstract form. Thus, we believe that the concept of path should be taught explicitly,

that the concept of path can be used to organize beginning geometric notions, and that appropriately connected physical and Logo activities offer an ideal environment for studying paths and related geometric notions.

The Logo philosophy and the constructivist philosophy of the curriculum standards have the same two major goals (Clements & Battista, 1990). First, students should actively experience building ideas and solving personally meaningful problems. Second, students should become autonomous and self-motivated. Taylor (1980) distinguish the traditional geometry curriculum from a Logo geometry curriculum by stating that children who are engaged in Logo activities will invent basic concepts in mathematics, thereby learning "to be mathematicians" versus learning "about mathematics." Furthermore, Logo based geometry activities can promote substantive rather than factual learning, helping students progress to higher levels of thinking in geometry.

As an example, consider the concept of rectangle. In the usual elementary school geometry curriculum, students are required only to be able to identify a visually presented rectangle--a level-1, visual activity in the Van Hiele hierarchy. In Logo, however, students can be asked to construct a sequence of commands, a procedure, to draw a rectangle. This process forces them to make their concept of rectangle explicit. They must analyze the visual characteristics of the rectangle and establish relations among its component parts. For example, students who think of a rectangle as "a figure with two long sides and two short sides" must be more precise and complete to write a Logo procedure for a rectangle; they must explicitly address properties of rectangles, such as opposite sides being equal in length and adjacent sides being perpendicular. For instance, to draw 80 unit by 40 unit rectangle, students have to apply numbers to the measures of the sides and angles, or turns (Figure 1). This process helps them become explicitly aware of such characteristics as "opposite sides equal in length." If instead of fd 60 they enter FD 90, the figure will not be a rectangle. The link between the symbols and the figure is direct and immediate. Studies confirm that students' ideas about shapes are more mathematical and precise after using Logo (Clements & Battista 1989, 1992).

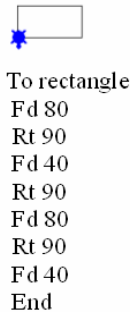


Figure 1. A procedure to construct a 40 by 80 unit rectangle

RESEARCH ON LOGO

Many Logo projects have attempted to explore the benefits of Logo programming for mathematics learning. In most, the instructional focus has been on Logo as a programming language and environment for exploration. Evaluation of these projects has indicated that this approach to increasing mathematics achievement is generally ineffective (Akdag, 1985; Blumenthal, 1986). However, it is possible that the students in these projects learned concepts that were not part of the standard curriculum and thus were not assessed, or that their teachers did not lead them to see the connections between the Logo-based concepts and other mathematical tasks.

In a few studies that attempted to make connections between students' work with Logo and textbook it was found significant increases on tests of geometric achievement (Howe, O'Shea, & Plane 1980; Lehrer & Smith, 1986).

Proper Logo environments may help students make the transition from the visual to the descriptive level of thought in the Van Hiele hierarchy. In fact, after working with the Logo activities, students attempting geometric tasks were less likely to conceptualize shapes on the basis of their visual appearance, and more likely to conceptualize them in terms of their properties (Battista & Clements, 1988).

Some students understand certain ideas, such as angle measure, for the first time only after they have used Logo. They have to make sense of what it is being controlled by the numbers they give to right- and left-turn commands. The turtle immediately links the symbolic command to a sensory-concrete turning action. Receiving feedback from their explorations over several tasks, they develop an awareness of these quantities and the geometric ideas of angle and rotation (Kieran & Hillel, 1990).

OUR EXPERIENCES WITH LOGOTURK

There are some difficulties on using Logo. Although Logo commands are in intuitive nature, non-English users may have difficulties on the usage of these commands. WinLogo has been partly used in some computer related courses in Faculty of the Education at Abant Izzet Baysal University in the spring semester year of 2004. WinLogo was chosen as our Logo platform since it was the windows version since it was free and could be downloaded form Internet. Firstly, basic Logo commands were introduced to students and they were instructed on the rationale behind how to use these commands to construct complex geometric shapes. Since commands were in English, students were required to memorize the commands. After students made some practice on basic Logo commands such as moving turtle one point to another, they were given instruction on how to incorporate these commands together to construct certain geometric figures. They were given a task of drawing a simple house by the help of the basic logo commands. Although most students were able to draw a simple house at the end of certain time period, the way they produced the houses suggested that they had difficulty of grasping the working mechanism of the Logo. They mostly preferred to imitate instructor’s way of drawing a house instead of trying out other possibilities for complex house structures. The Figure 2 shows a simple house most students were able to design.

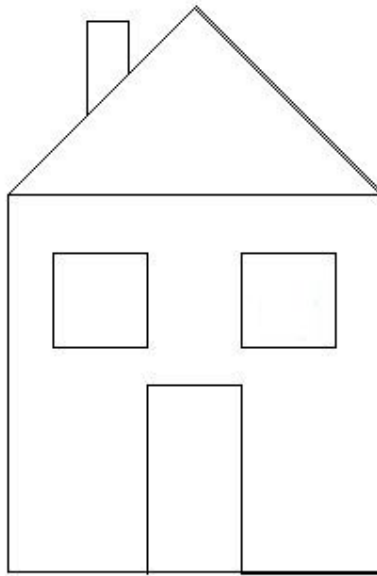


Figure 2. A house figure created with Logo commands

After several interviews and our observations with the students, it was discovered that Logo commands being in English created a big hurdle for students to internalize Logo commands. Furthermore, two separate windows of Winlogo interface designed for editing procedures and running them were not seen appropriate for Turkish students since it is not found user-friendly. Most students had difficulties in saving their work and lost their work since using save command in the main window did not save the program content in the edit window. Winlogo Editing window required procedures to be entered within a “to” and “end” block. Some students also mistakenly deleted the procedure name after “to”. Hence, procedures were not defined properly although the correct command sequences were typed. Moreover, simple typing errors on writing commands such as omitting to put a space between two commands or a command and its argument or misspelling an English word caused problems. Not being able to save the procedures correctly was the most significant problem as mentioned above. The latest mode of the program, namely pen’s being down or the elements already drawn on the screen, also caused confusion. Although some students were able to draw a house, it became apparent later that they entered commands line by line instead of writing them inside a procedure. This has lead to not being able to re-produce what has been done in another computer since it was not saved properly. Thus, a Turkish version of the Logo was designed and LogoTurk was developed by Erol KARAKIRIK and Soner DURMUS at the later stage of the semester.

The main concern of Turkish design is to minimize Turkish students’ problems with the interface and direct them to more conceptually favorable experiences with Logo commands by eliminating the language barrier, since students’ main problem is language barrier. Main goal is to eliminate language barrier and take adVantage

of conceptual part of original Logo. Proposed Logo (LogoTurk) has considered these difficulties. It is designed in a way that commands can be entered in Turkish. This has enabled students to use their own language to move the turtle in any direction they want. For example, instead of using “fd” command, students could use “ig”. Students can use English, Turkish or combination of both in LogoTurk. LogoTurk design included both Turkish and English commands in both long and short forms. Table 1 shows LogoTurk commands that are all equivalent and could be used interchangeable.

Table 1. Some LogoTurk commands

Explanation	Turkish Short	Turkish Long	English Short	English Long
Moves turtle 40 Pixels Forward	ig 40	İleri 40	Fd 40	Forward 40
Moves turtle 40 Pixels backward	gg 40	Geri 40	Bk 40	Back 30
Set the pen mode down, it paints	Ka	kalemaşağı	pd	Pendown
Set the pen mode up, it does not paint	Ky	kalemyukarı	pu	Penup

All forms of all commands were implemented in LogoTurk. One can easily change from one mode to another by related language menu commands. For instance, all commands could be change from Turkish short form to English long form or from English long form to Turkish short form. Screenshots in Figure 3 show four different modes of the program for the same commands.

This language feature is aimed to teach students English versions of the commands in order to be used in classical Logo packages if needed. Using Turkish equivalent of basic commands seemed to relieve students’ anxiety towards Logo. Hence, it is suggested that English version of Logo should not be seen as a standard and the development of local versions of logo should be supported.

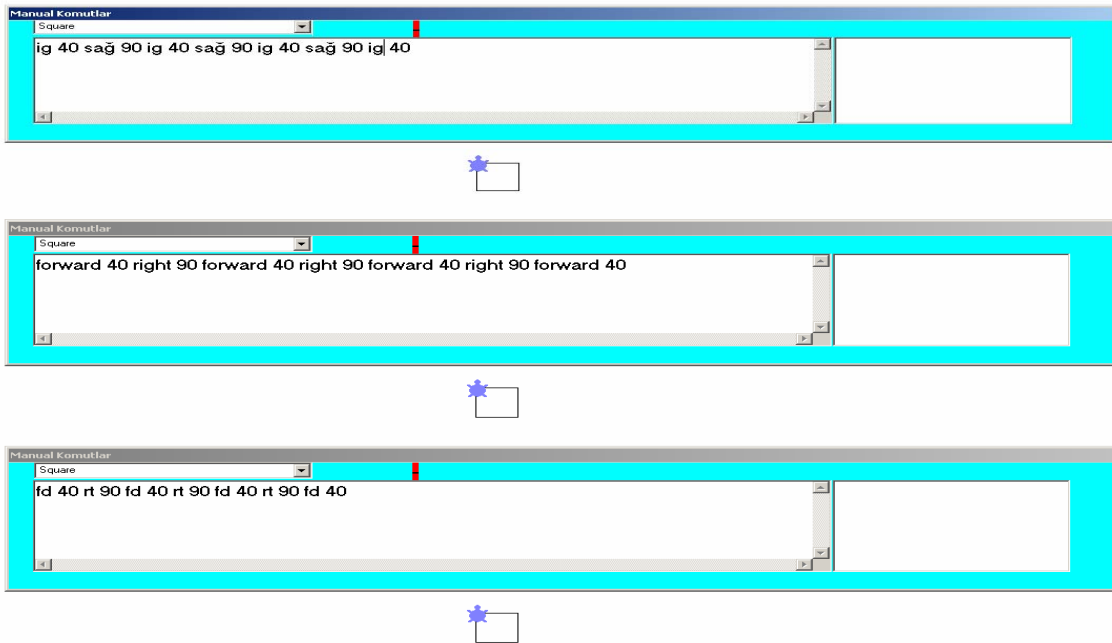


Figure 3. Screenshots from LogoTurk showing four different modes for the same commands

Taking students’ difficulties into account, it is also decided to put editing window inside the main window. It was observed that students had fewer difficulties with this sort of interface design than WinLogo. In classical Logo packages, a procedure starts with “to” and ends with “end”. All commands of a procedure are between “to” and “end”. The procedures’ name must be entered after “to”. All commands can be written in one line by separating them with a space. For example, to construct a square, a procedure could be written as follows:

To square
fd 40 rt 90 fd 40 rt 90 fd 40 rt 90 fd 40
end

Or

To square
repeat 4 [fd 40 rt 90]
end

To run a procedure, students need to select all lines of commands and select Test on toolbar. If students want to use a procedure on main screen, they need to enter name of the procedure. Each procedure can be tested separately on procedure screen. Main problem with using procedures is that students need to save each procedure to use on main screen even if a procedure screen is open.

LogoTurk has a different approach to procedure definition. Although, LogoTurk supports classical procedure definition, the programmatic nature of the procedure definition were left and “to” and “end” block was eliminated. Instead, a procedure combo box was employed in order to be able to move among procedures. The procedure combo box eliminated some of the problems students experienced with Winlogo. The simplification of procedure definition has lead students defining several different procedures instead of putting all the commands in a procedure. Furthermore, “save command” was implemented in a way that all the procedures students working on were saved at the same time. This has minimized the students’ loss of data. This design also enabled testing of each procedure separately by the help of several commands. It is possible to add, delete, rename, run and stop each procedure separately by related menu items and shortcuts in LogoTurk.

LogoTurk has a “MainScreen” procedure by default. It was considered as the main procedure of the program. However, one could rename this procedure or assign another procedure as the main procedure if needed. Several procedures can be written on procedure screen and any procedure could be used in another procedure. Recursive procedure definition is also accepted. However, this feature should be used with caution in order not to put the program into infinite loop or not to crash the program. Procedures can also be defined with arguments. Table 2 shows a procedure that defines a square with a varying length meaning with an argument.

Table 2. Examples of procedures in LogoTurk

<i>To square _length</i> <i>Repeat 4 [fd _length rt 90]</i> <i>end</i>	Square 40 draws a square with length 40 unit Square 100 draws a square with length 100 unit Square 40*sin(45) draws a square with length 40*sin(45) unit
<i>To polygon _length _side</i> <i>Repeat _side [fd _length rt 360/_side]</i> <i>end</i>	Polygon 40, 4 draws a square with side length 40 unit Polygon 100, 8 draws a hexagon with side length 100 unit

It is also possible to use arithmetic operations and mathematical functions while entering the arguments of the functions. This enables to define more complex functions. For instance, instead of defining square and hexagon separately, a regular polygon of any number of sides could be defined by two arguments, namely length and the number of sides. In order to draw a regular polygon, the number of repetition to make is equal to number of sides of the polygon and the angle measure of each turn could be found by dividing 360 to number of sides as shown above.

Another advantage of LogoTurk is its ability to detect simple typing errors and automatically correct them. Any simple typing error on writing commands can cause a program not executing the command or procedure which might be crucial part of the intended end figure. For instance, to move the turtle 40 unit forward, students need to enter Forward 40. If one forgets to put space between Forward and 40, this command is not going to be executed. Classical Logo shows an error message but does not show where the errors come from and ignore the errors. However, this might confuse students since they expect those commands to be executed and not knowing why the commands are erroneous and not getting any feedback and correction of the errors is not educationally favorable.

LogoTurk also suggests a solution for this confusion. The difficulty of any program of detecting possible typing errors is very important for users to know where errors begin and end. High level programming languages, such as C++, try to overcome this problem by requiring users to enter semicolon”;” after each separate command. Hence, this approach was adapted for LogoTurk design. LogoTurk could easily detect the aforementioned typing errors if semicolons are put after every command. However, when spaces are used to separate commands LogoTurk has difficulty of detecting this sort of errors although it could also detect some typing errors of this sort. If every command is written on a separate line, LogoTurk could always detect typing errors related to space usage whether or not a semicolon used or not. Hence, students can write commands without worrying about putting space between commands. LogoTurk can recognize erroneous commands in both English and Turkish. For example, command “ fd50” is same as the command “fd 50”. Similarly, command “ig50” is same as the

command “ig 50”. Any mistake in writings is being detected and corrected if “auto correct” mode was selected. If the “stop on error” option was selected, the program execution stops after encountering a typing error. LogoTurk places the cursor to the first error encountered during execution after executing all commands. This feature has helped students on detecting their own errors and focusing on analyzing different attributes of figures (Figure 4).

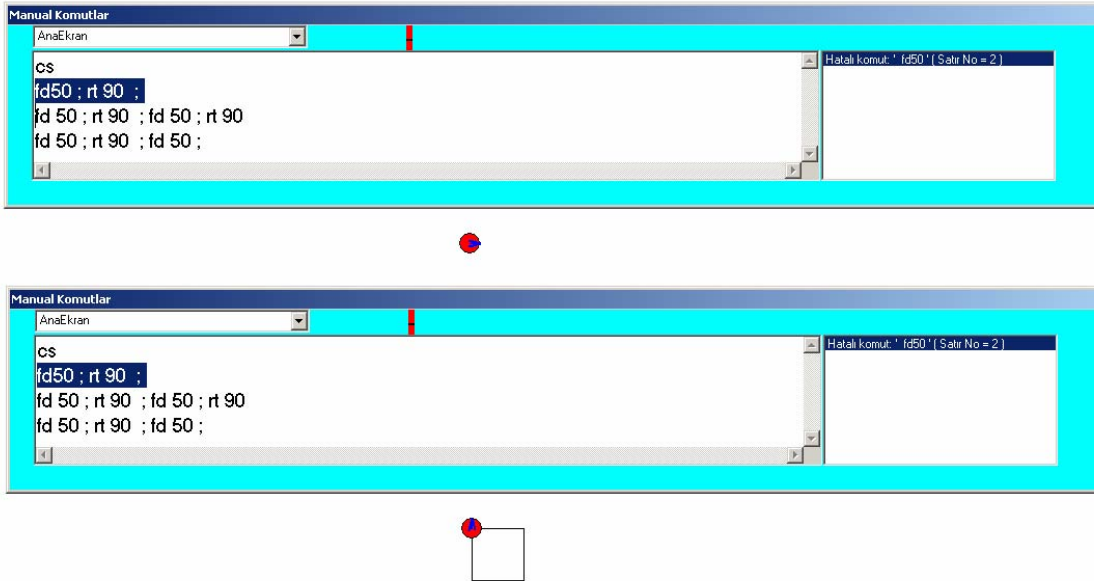


Figure 4. Screenshots from LogoTurk showing errors.

In addition, LogoTurk displays information about the current state of the program at the top of the screen with the help of a statusbar including the current position of the turtle, whether the pen is down or up, the direction of the turtle, the pen color, screen color and fill color, the current location of the mouse and the current position of home. The status bar (Figure 5) can also be hidden if needed. In classical Logo, these type of information can be obtained by writing some commands including Print ?, print xcor, print ycor, etc.



Figure 5. A view of Status bar in LogoTurk

Other classical Logo commands related to making loops and conditional executions are also included in LogoTurk but they are out of scope of this paper.

Students felt much more comfortable with LogoTurk than WinLogo because of the differences outlined in this paper. After getting used to LogoTurk, students begin to enjoy their Logo experiences and they were able to produce several complex figures such as houses, cars etc. Figure 5 shows a complex house students could produce after LogoTurk was employed in the course. Another example is given in Appendix A with necessary LogoTurk commands. Appendix A has also commonly used LogoTurk commands.

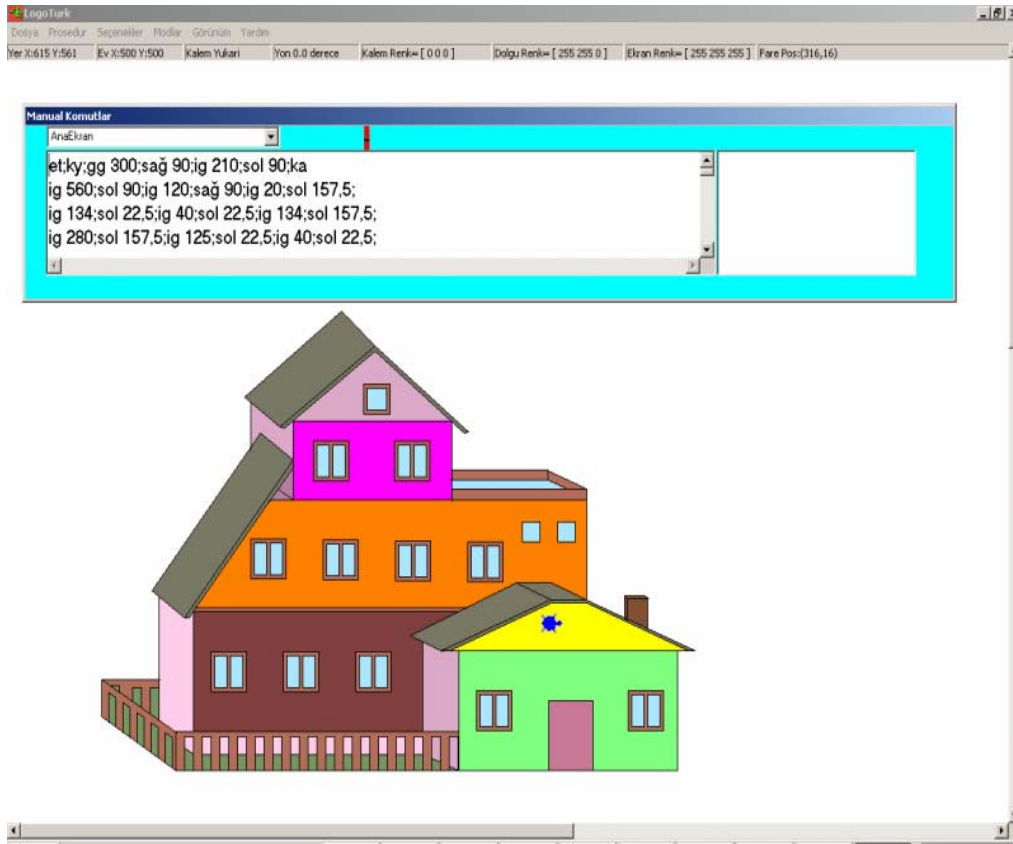


Figure 4. A house figure created with LogoTurk commands.

RECOMMENDATIONS FOR FUTURE RESEARCH

It is proposed that designing user friendly interfaces for Logo may change dramatically students' perception of Logo and made them focus on more conceptual oriented geometrical tasks. There is still room for making changes in the graphical interface of LogoTurk to have students much better experiences and interactions with Logo geometry. It is proposed that LogoTurk enhanced Turkish students' Logo geometry experiences by removing the English language barrier. Requiring students to type Logo commands line by line is not seen proper in Windows environment. Hence, there is need for improvement in graphical interface. It is suggested that replacing some basic commands of Logo with user-friendly graphical elements might improve students' interaction and help them understand better the geometrical concepts and relations.

There is also a need to investigate the effects of a Logo-based learning environment on students' attitudes and problem solving performances at the primary grades in general towards geometry. Logo environments have the potential to transform both the method and content of the elementary geometry curriculum.

Our experience with LogoTurk suggests that Logo-based learning environments have the potential to enhance students' geometry experiences and facilitate constructing their geometrical concepts and relations. This might also have consequences for changing the methods and the content of the current elementary geometry curriculum.

REFERENCES

- Akdag, F. S. (1985). *The effects of computer programming on young children's learning*. Unpublished doctoral dissertation, Ohio State University.
- Battista, Michael T. (2002). Learning Geometry in a Dynamic Computer Environment. *Teaching Children Mathematics*, 8 (6): 633-639.
- Battista, M. T., & Clements, D. H. (1988). A case for a Logo-based elementary school geometry curriculum. *Arithmetic Teacher*, 36: 11-17.
- Blumenthal, Wendy. (1986). *The Effects of Computer Instruction on Low Achieving Children's Academic Self-beliefs and Performance*. Unpublished doctoral dissertation, Nova University.

- 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.
- 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.
- Clements, D. H., & McMillen, S. (2001). *Logo and Geometry*. *Journal for Research in Mathematics Education Monograph Series*, Arlington, VA: National Science Foundation.
- Clements, D. H., & McMillen, S. (1996). Rethinking Concrete Manipulatives. *Teaching Children Mathematics*, 2(5), 270-279.
- 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.
- Clements, Douglas H., & Battista, Michael T. (1990). Research into Practice: Constructivist Learning and Teaching." *Arithmetic Teacher*, 38: 34-35.
- Clements, Douglas H., & Michael T. Battista. (1989). Learning of Geometric Concepts in a Logo Environment. *Journal for Research in Mathematics Education* 20: 450-67.
- Clements, Douglas H., & Battista, Michael T. (1986). Geometry and Geometric Measurement. *The Arithmetic Teacher*, 33: 29-32.
- Driscoll, Mark J. (1983). *Research within Reach: Elementary School Mathematics and Reading*. St. Louis: CEMREL.
- 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.
- 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.
- Kouba, V. L., Brown, C. A., Carpenter, T. P., Lindquisti M. M., Silver, E. A., & Swafford, J. O. (1988). Results of the fourth NAEP assessment of mathematics: Measurement, geometry, data interpretation, attitudes, and other topics. *Arithmetic Teacher*, 35(9): 10-16.
- Lehrer, R., & Smith, P. C. (1986). *Logo learning: Are two heads better than one?* Paper presented at the meeting of the American Educational Research Association, San Francisco, CA.
- The International Study Center. (1999). *TIMMS report*. This report is available at <http://timss.bc.edu/timss1999.html>.
- National Council of Teachers of Mathematics. (2000). *Principles and Standards for School Mathematics*, Reston: The National Council of Teachers of Mathematics, Inc.
- Papert, Seymour. (1980). *Mindstorms*. New York: Basic Books.
- 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.
- Sowell, Evelyn J. (1989). Effects of Manipulative Materials in Mathematics Instruction. *Journal for Research in Mathematics Education*, 20: 498-505.
- Stigler, J. W., Lee, S. Y. & Stevenson, H. W. (1990). *Mathematical knowledge of Japanese, Chinese, and American elementary school children*. New York: The Free Press.
- Suydam, Marilyn N. (1986). Research Report: Manipulative Materials and Achievement. *Arithmetic Teacher*, 10: 32-32.
- Taylor, Robert P. (Ed.). (1980). *The Computer in the School: Tutor, Tool, Tutee*, 177-96. New York: Teacher's College Pres.
- Trafton, Paul R., & LeBlanc, John F. (1973). Informal Geometry in Grades K-6." In Kenneth B. Henderson (ed.), *Geometry in the Mathematics Curriculum*, Thirty-sixth Yearbook of the National Council of Teachers of Mathematics, 11-51. Reston, Va.: The National Council of Teachers of Mathematics, Inc.

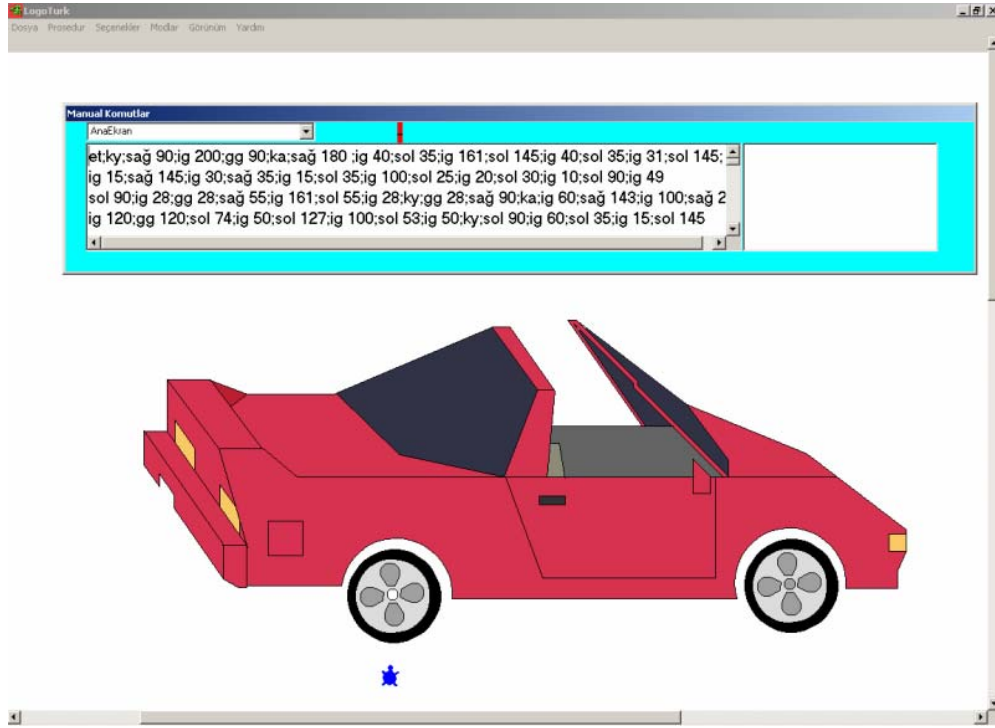
APPENDIX A

A car drawn in LogoTurk

Procedure AnaEkran

et;ky;sağ 90;ig 200;gg 90;ka;sağ 180 ; ig 40;sol 35;ig 161;sol 145;ig 40;sol 35;ig 31;sol 145;
ig 15;sağ 145;ig 30;sağ 35;ig 15;sol 35;ig 100 ;sol 25;ig 20;sol 30;ig 10;sol 90;ig 49
sol 90;ig 28;gg 28;sağ 55;ig 161;sol 55;ig 28;ky ;gg 28;sağ 90;ka;ig 60;sağ 143;ig 100;sağ 21
ig 120;gg 120;sol 74;ig 50;sol 127;ig 100;sol 53;ig 50;ky;sol 90;ig 60;sol 35;
ig 15;sol 145; ka;ig 25;sağ 145;ig 40;sağ 35;ig 25;sol 35;ky
ig 51;sol 145;ka;ig 25;sağ 145;ig 31;ky;ig 9 ;sağ 35;ig 8;ka;ig 18;ky;gg 179;sağ 90;ig 34
sol 160;ka;ig 46;sol 20;ig 100;sol 23;ig 200 ;sağ 23;ig 20;sağ 55;ig 90;sağ 40;ig 100;sağ 85
ig 50;sağ 111;ig 108;sağ 70;ig 20;ky;gg 20;ka ;sol 127;ig 90;ky;sol 149;ig 175;sağ 95;ka;ig 240
sağ 40;ig 53;ky;sağ 13;ig 60;sağ 90;ka;ig 25;ky ;sağ 36;ig 102;sağ 90;ig 2;sol 45;ka;ig 102
sol 32;ig 125;sol 12;ig 220;ky;ig 20;ka;ig 20; sol 135;ig 150;sağ 45;ig 5;sol 45;ig 100;
sol 45;ig 10;sol 125;ig 150;sol 55;ig 37;ky; gg 33;sol 125;ka;ig 145;ka; sağ 170;ig 98;ky
sağ 45;ka;ig 5;sol 45;ig 147;sol 45;ig 6;sol 90; ig 20;sol 37;ig 80;sol 15,5;ig 160;ky;gg 160;
sağ 165;ka;ig 150;sağ 15;ig 145;sağ 52;ig 25; sağ 90;ig 20;sağ 90;ig 20;sağ 90;ig 20;sağ 90;
ky;ig 20;sağ 23;ka;ig 25;sol 23;ig 20;sağ 90,5;ig 60;sağ 90 ;cember 9,8;
sağ 110;ig 330;sağ 90; cember 9,8 ;sağ 86 ;ig 110;ky;sağ 90;ig 35;sağ 90;ig 25
ka;ig 40;sol 90;ig 40;sol 90;ig 40;sol 90;ig 40;ky ;gg 91;sol 90;ig 275;sağ 70;ka;ig 125;sol 70
ig 200;sol 90;ig 117;ky;sol 90;ig 5;sağ 45;ka;ig 30 ;sol 135;ig 40;sol 90;ig 20;sol 90;ig 20;ky;ig 58;sol 90
ig 80;ka;ig 105;ky;ig 5;sol 90;ig 20;sol 90;ig 5; ka;ig 10;sağ 80;ig 42;ky;sağ 10;ig 20;ka
ig 10;sağ 90;ig 30;sağ 90;ig 10;sağ 90;ig 30 ;ky;sol 90;ig 22;sağ 90;ig 185;ka;ig 130;ky
sağ 90;ig 130;sol 90;ky;gg 55 ;teker ;ky;sol 68;ig 67;sağ 90;ka;ky;ig 445;sağ 180
teker ;ky;sağ 90;ig 40;degdr 215,50,80;doldur; sol 80;ig 80;degdr 215,50,80;doldur ;ig 80;degdr
215,50,80;doldur
sağ 90;ig 8;degdr 215,50,80;doldur ;sol 90;ig 15;degdr 215,50,80;doldur ; gg 32;degdr 215,50,80;doldur ;sağ
90;gg 3;degdr 215,50,80;doldur
sol 10;ig 25;degdr 250,200,100;doldur; ig 60;degdr 215,50,80;doldur ;sol 25;ig 50;degdr 250,200,100;doldur
sağ 90;ig 50;degdr 215,50,80;doldur ;ig 30;degdr 190,30,50;doldur;sağ 80;ig 50;degdr 215,50,80;doldur; sol
40;ig 150;degdr 50,50,70;doldur
sağ 40;ig 170;degdr 215,50,80;doldur; ig 40;degdr 50,50,50;doldur;sol 130;ig 40;degdr 215,50,80;doldur; sağ
5;ig 100;degdr 215,50,80;doldur; sağ 90;ig 100;degdr 215,50,80;doldur; gg 15;degdr 50,50,70;doldur ; ig
30;degdr 50,50,70;doldur ;sağ 140;ig 145;degdr 140,140,120;doldur ;gg 40;degdr 100,100,100;doldur;sol 115;ig
155;degdr 215,50,80;doldur; sol 80;ig 60;degdr 215,50,80;doldur ;sağ 85;ig 222;degdr 250,200,100;doldur ;sağ
120;ig 92;degdr 0,0,0;doldur; ig 8;degdr 220,220,220;doldur ;ig 8;degdr 160,160,160;doldur ;ig 30;degdr
160,160,160;doldur; sağ 70;ig 35;degdr 160,160,160;doldur ;sağ 80;ig 30;degdr 160,160,160;doldur ;sol 35;gg
35;degdr 160,160,160;doldur ;sol 85;ig 405;degdr 0,0,0;doldur;ig 15;degdr 220,220,220;doldur;sağ 5;ig
15;degdr 160,160,160;doldur; ig 40;degdr 160,160,160;doldur ;sağ 135;ig 25;degdr 160,160,160;doldur; sol
45;gg 35;degdr 160,160,160;doldur;gg 70;

Procedure cember_a_b tekrar 25[ig_a;sol_b]
Procedure cember_c_d tekrar 22,9 [ig_c;sol_d]
Procedure teker ka;tekrar 36[ig 1;sol 10];sk;gg 4;ky;sağ 90;ig 4;sağ 22;ka;ig 10; tekrar 15[ig 3;sol 15] ;sol
10;ig 14;sol 55;ig 10 ;ky;gg 15;sağ 90;ig 7;sağ 90; sağ 22;ka;ig 10;tekrar 15[ig 3;sol 15] ;sol 10;ig 14;sol 55;ig
10;gk;ky ;gg 15;sağ 80;ig 7;sağ 95; sağ 22;ka;ig 10 ;tekrar 15[ig 3;sol 15] ;sol 10;ig 14;sol 55;ig 10;sk ;gk;ky;gg
15;sağ 90;ig 7;sağ 90 ;sağ 22;ka;ig 10 ;tekrar 15[ig 3;sol 15] ;sol 10;ig 14;sol 55;ig 10;sk;gk;sol 5;ky;ig 38; sol
90;gg 12;ka ;tekrar 46[ig 6;sol 8]; ky;sağ 120;ig 12;sol 117;ka ;tekrar 50[ig 7; sol 7,5]



Commonly Used LogoTurk Commands

English Long	English Short	Turkish Short	Turkish Long	Turkish Long
ClearScreen	CS	ET	EkranTemizle	Clears the Screen
CLEAN	CLEAN	Temizle	Temizle	Clears the Screen
FORWARD ?	FD ?	İg ?	İleri ?	Moves Turtle ? pixel forward
BACK ?	BK ?	Gg ?	Geri ?	Moves Turtle ? pixel backward
RIGHT ?	RT ?	Sağ ?	Sağ ?	Turns Turtle's direction ? angle to right
LEFT ?	LT ?	Sol ?	Sol ?	Turns Turtle's direction ? angle to left
REPEAT ? []	REPEAT ? []	Tekrar ? []	Tekrar ? []	Repeats the command block ? times
HIDE TURTLE	HT	SK	SaklaKaplumbağa	Hides the turtle figure
SHOW TURTLE	ST	GK	GösterKaplumbağa	Show the turtle figure
PEN UP	PU	KY	KalemYukarı	Set pen up - it does not paint
PEN DOWN	PD	KA	KalemAşağı	Set pen down - it paints
PEN PAINT	PPT	Kyaz	KalemYaz	Set pen writing mode to paint
PEN ERASE	PE	Ksil	KalemSil	Set pen writing mode to erase
PEN REVERSE	PX	Kters	KalemTers	Reverse the pen writing mode
SETPENCOLOR[? ? ?]	SETPC [? ? ?]	Degkr	DeğiştirKalemRenk	Change turtle's pen color
SETSCREENCOLOR[? ? ?]	SETSC [? ? ?]	Deger	DeğiştirEkranRenk	Change screen's background color
SETFLOODCOLOR [? ? ?]	SETFC [? ? ?]	Degdr	DeğiştirDoldurRenk	Change filling color
FILL	FILL	Doldur	Doldur	Fills a closed region with fill color
SETX ?	SETX ?	KurX	KurX	Set the new x coordinate of the

				turtle
SETY ?	SETY ?	KurY	KurY	Set the new y coordinate of the turtle
SETPOS [? ?]	SETPOS [? ?]	KurPos	KurPos	Set the new x,y coordinates of the turtle
HOME	HOME	Ev	Ev	Set the turtle to initial position
LABEL [?]	LABEL [?]	Yaz ?	Yaz ?	Insert a text to turtle's current position
PRINT HEADING	PRINT HEADING	YazY	YazYön	Prints Turtle's direction
SET HEADING ?	SetH ?	SetY	SetY	Set turtle's new direction to ?angle
WAIT ?	WAIT ?	Bekle ?	Bekle ?	Wait for ?/60 seconds before next command
SETPENWIDTH	SETPW	KK	KalemKalınlık	Sets the pen width
FONTSIZE ?	FS ?	YazıBoyuu ?	YB ?	Sets the font size for text input
FONTNAME ?	Fn ?	YazıAdı ?	YA ?	Sets the font name for text input
STOP	STOP	DURDUR	DURDUR	Stops the execution of a procedure
RESET	RESET	SIFIRLA	SIFIRLA	Resets alll paramters of the program
IFTRUE	IFTRUE	EĞERDOĞRU	EĞERDOĞRU	Executes if true
IFFALSE	IFFALSE	EĞERYANLIŞ	EĞERYANLIŞ	Executes if false
UNTIL	UNTIL	KADAR	KADAR	Executes till cetain conditions are met
WHILE	WHILE	İKEN	İKEN	Executes while conditions are met
FOR	FOR	DÖNGÜ	DÖNGÜ	Executes a loop