



# Educational Technology Lab

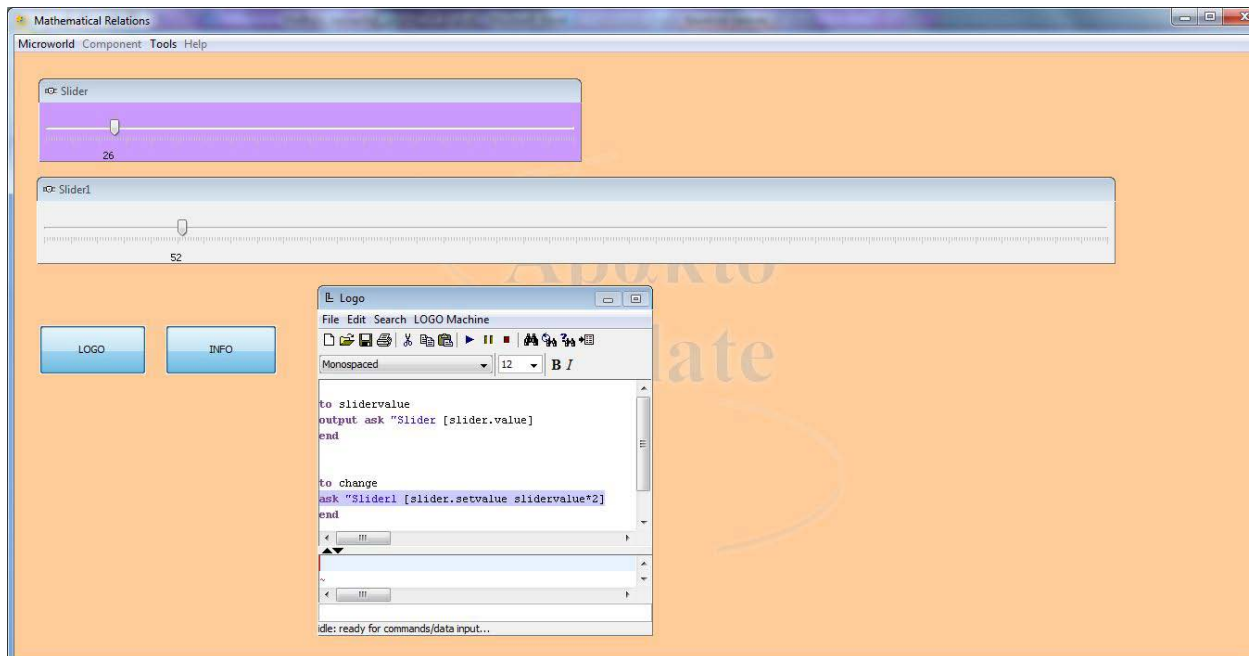
National and Kapodistrian University of Athens

School of Philosophy

Faculty of Philosophy, Pedagogy and Philosophy (P.P.P.),

Department of Pedagogy

*Director: Prof. C. Kynigos*



## "MaRs" Manual

---

## Table of Contents

1. The MaRs Microworld.....	3
1.1 Introduction .....	3
1.2 Description of MaRs.....	3
1.3 Structure and function of the sliders .....	4
1.4 Sliders' Connections.....	4
2. Modifying the MaRs microworld .....	6
2.1 Adding an extra slider .....	6
2.2 Slider settings.....	6
2.3 Logo code modification .....	8
2.3.1 Examples of Slider connections through Logo code.....	9
Appendix.....	11

# 1. The MaRs Microworld

## 1.1 Introduction

There are some mathematical concepts, such as the variable and the function, that have been proven to contain high levels of abstraction in their definition. This fact makes it difficult for students to comprehend them. In addition, the teaching of these concepts is usually limited to solving standard mathematical exercises and to memorizing formulas. However, in that way students don't acquire profound knowledge of the concepts, but superficial knowledge of some rules and definitions. The method that is usually used for teaching the functions is filling in tables with the function values and solving common mathematical problems. With the MaRs microworld, we suggest a different representation of the function concept. The MaRs microworld projects a more descriptive representation, which challenges the student to explore and be actively engaged in the cognitive process of comprehending the specific concept. The main aim of the microworld is to support the student in discovering the mathematical relation that connects two (or more) sliders, according to which the value of the one slider depends on the value of the other.

## 1.2 Description of MaRs

The MaRs kit was developed on the E-slate platform, an environment that enhances Inquiry-based learning and enables the user to construct microworlds aiming at experimentation and exploration of phenomena and concepts. MaRs is a microworld as well, consisted of the following components:

- 2 Sliders placed one above the other, with the names "Slider" (basic slider) and "Slider1".
- The Logo Editor component: It contains the Logo code of the microworld which controls the behavior of all the other components (e.g. the Sliders).
- 1 "INFO" button which displays/hides an informative text about the operation of the microworld.
- 1 "LOGO" button which displays/hides the Logo Editor Component.

MaRs is a model microworld which can be easily modified by the instructor so that it is adjusted to his

needs. In the next unit, there is a detailed description of the function of the microworld components as well as the process to be followed for the microworld modification.

### 1.3 Structure and function of the sliders

The two sliders of the microworld are connected through a mathematical relation. This means that when the user moves the cursor of the first slider to a number, the cursor of the second slider will automatically be moved to the number which is the result of this mathematical relation. For instance, if the mathematical relation is  $Slider1 = Slider + 5$  and the *Slider* points at the value 3, then the *Slider1* will point at the value 8, etc. Therefore, each slider represents a different variable of the same mathematical relation. This creates a model of a linear representation of mathematical functions - instead of the usage of traditional tables of values - which can be used in a variety of ways in algebra field.

In this model microworld, we used the mathematical relation "***Slider1 = Slider \* 2***" by default.

### 1.4 Sliders' Connection

The connection between the 2 sliders is programmed through some Logo commands, which can be displayed by clicking on the "LOGO" button. In particular, the mathematical relation that connects the two sliders is defined in this line of the code:

```
ask "Slider1 [slider.setvalue slidervalue*2]
```

The command *slider.setvalue* makes the *Slider1* point at the value that is written right after the command. For example, if we write `ask "Slider1 [slider.setvalue 5]`, then the *Slider1* will point at the value 5. In our case, the value that we set for the *Slider1* is the result of the mathematical relation ***slidervalue\*2*** (where *slidervalue* is the process which 'takes up' the current value of the basic *Slider*).

To sum up, the above command consists of the following parts:

```
ask "Slider1 [slider.setvalue slidervalue * 2]
```

Diagram illustrating the components of the command:

- Slider1**: Name of the slider
- slider.setvalue**: Logo command for setting the value of the slider
- slidervalue \* 2**: Mathematical relation

So, if, for example, we wanted to connect the sliders with the relation  $y = x + 3$  or  $\text{Slider1} = \text{Slider} + 3$  we would change the command to:

*ask "Slider1 [slider.setvalue slidervalue + 3]*

## 2. Modifying the MaRs microworld

The MaRs model microworld enables the user to modify it through simple procedures and have full control of its sliders, as well. The MaRs environment provides many possibilities of modification and expansion, which are described below.

### 2.1 Adding an extra slider

To **add** a new slider, follow this path from the horizontal menu at the top of the window:

**Component -> New -> User Interface -> Slider**

and a new slider will be added to the microworld.

To **delete** a slider from the microworld, right click on the control bar of the slider's window and then click on the option "Remove" of the popup menu.

### 2.2 Slider settings

To **rename** a slider, right click on the control bar of the slider's window and then click on the option "Rename". Insert the new name at the popup window and click "OK".

To change the height and the width of a slider, adjust the size of its window by clicking on its edge and dragging it with the mouse.

To display and modify other settings of a slider, select from the horizontal menu at the top of the microworld's window "**Tools -> Component Editor**". This will display the "Component Editor" at the left side of the screen (Fig. 1). At the top of this window, you can select the component you want. Once you select a component, its settings will be displayed below. Some important setting fields of the slider component are the following:

- Major tick spacing: It sets the space between the large scale ticks of the slider. If it is set to the value 0 the large ticks will not be displayed.
- Minor tick spacing: It sets the space between the small scale ticks of the slider. If it is set to the value 0 the small ticks will not be displayed.

- Minimum: It sets the slider's minimum value.
- Maximum: It sets the slider's maximum value.
- Orientation: It sets the slider's orientation (Horizontal or Vertical).
- Font: It sets the font and the size of the slider's numbers.
- Paint Ticks: It defines whether scale ticks are displayed on the slider.
- Paint Labels: It defines whether the numbers of the scale are displayed on the slider.
- Paint Track: It defines whether the slider track is displayed.
- Text Color: It sets the color of the slider numbers
- Background: It sets the slider background color.
- Inverted Value: It inverts the slider ends.

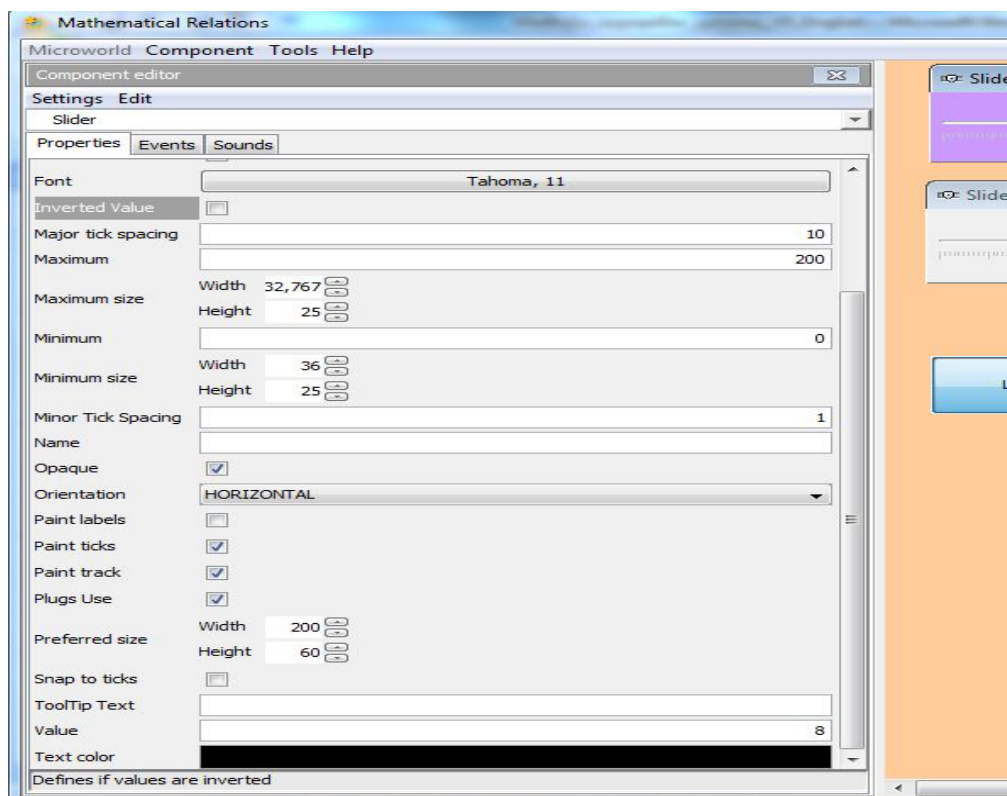


Figure 1: Settings of the 'Slider' component

### 2.3 Logo code modification

The Logo code of MaRs, gives you the possibility to control the sliders' behavior. The Logo editor is displayed on the screen, when you click on the "LOGO" button. The default code contains two processes. The first one takes up the basic slider's current value and the second one sets the relevant value to the second slider.

The process that takes up the basic slider's current value ("*Slider*") is the *slidervalue* and is comprised of the following commands:

```
to slidervalue
  output ask "Slider [slider.value]
end
```

The outcome of the above commands is that every time we write *slidervalue* in our code, the above process is run and it 'returns' the current value of the *Slider*.

The process *change* that sets the relevant value to the second slider is comprised of the following commands:

```
to change
  ask "Slider1 [slider.setvalue slidervalue + 2]
end
```

With the above process, the value of *Slider1* is set to *slidervalue + 2* which actually is *Slider + 2*.

You can edit this code and apply any change you want to it, as well. Some examples of code modifications are presented in the next unit.

In order to apply a new Logo code or a change in the existing code, you have to select with the mouse the code and click on the *Play* icon on the Logo editor's tool bar or the *insert* button of your keyboard. You can find the available Logo commands for this microworld in the [Appendix](#) of this manual.



### 2.3.1 Examples of Slider connections through Logo code

**Example 1:** 4 sliders. All three of them are connected with the main slider (“Slider”).

The mathematical relations are:

- Slider1 = Slider + 2
- Slider2 = Slider \* 2
- Slider3 = (Slider)<sup>2</sup>

The Logo code is:

```
to slidervalue
  output ask "Slider [slider.value]
end

to change
  ask "Slider1 [slider.setvalue slidervalue+2]
  ask "Slider2 [slider.setvalue slidervalue*2]
  ask "Slider3 [slider.setvalue power slidervalue 2]
end
```

**Note:** The command **power** (4<sup>th</sup> line of the process *change*) is a logo command that raises a number to a power and its syntax is: *power number power\_value*. Therefore, on the 4<sup>th</sup> line of the process *change* above, *ask "Slider3 [slider.setvalue power slidervalue 2]*, the command *power* raises the *slidervalue* of the basic slider to square.

**Example 2:** 3 Sliders. There is the basic slider (“Slider”), “Slider1” which is connected with the basic slider, and “Slider2” which is connected with “Slider1”. (Function Composition)

The mathematical relations are:

- Slider1 = Slider + 2
- Slider2 = Slider1 \* 2

The Logo code is:

```
to slidervalue
  output ask "Slider [slider.value]
```

*end*

*to change*

*ask "Slider1 [slider.setvalue slidervalue + 2]*

*end*

*to slidervalue1*

*output ask "Slider1 [slider.value]*

*end*

*to change1*

*ask "Slider2 [slider.setvalue slidervalue1 \* 2]*

*end*

As you can see from the examples above, the MaRs model microworld is open to many changes and modifications, so that the user is enabled to create his own learning environment, depending on the functions and the mathematical concepts he wants to teach.

# 1 Appendix

Logo commands for sliders' control:

Attention: Each command must be written inside the brackets of *ask "slider\_name [ ]*

<u>Command</u>	<u>Description</u>	<u>Example of use</u>
<b>Slider.Value</b>	Takes up the value pointed by the cursor of a slider	
<b>Slider.Setvalue <i>number</i></b>	Sets the cursor of a slider to point at the value of the <i>number</i>	Slider.Setvalue 10
<b>Slider.SetMax <i>number</i></b>	Sets the maximum value of a slider	Slider.SetMax 200
<b>Slider.SetMin <i>number</i></b>	Sets the minimum value of a slider	Slider.SetMin 200