Technology
Programming SmallBASIC

Gerard Thomas Labèrnia

preface

This document seeks to introduce students to the world of programming and is aimed at students from a level of second cycle of secondary or high school, the most appropriate technology in the curriculum is the 4th year of ESO. It can be used not only for technology, also drawing with other subjects, Mathematics, Physics, Music, English ....

At the time of the first commercial computer programs were rare, the software that is the machine software was developed by the users themselves, who often programmed their own applications. It is important to recover this spirit so that students realize the importance of knowing how to program a computer to do their own programs.

To simplify the learning of the matter has gone to directly enter the program code, allowing for a higher level by introducing algorithms each program.

The organization structure is subject tutorial. As a methodology recommended start

an explanation of the most important commands from the teacher, then move to the introduction of the examples for the students, store the disc and run each example. Therefore it is important to work with the publisher of the program. The student begins to make programs from the first day and see the results. We must also do the programming exercises, each student must store your disk and answer sheet when necessary.

Important note: to begin work is recommended that each student has a USB stick to store the files. You must install the program SmallBASIC 1 previously. You can download the installer program on the website: http://smallbasic.sf.net .

index

1. Introduction to SmallBASIC

Programming is the process of putting instructions on the computer to tell what functions are performed and in what order to do it. The aim of this subject began to make our first programs.

SmallBASIC is a language of computer programming based on BASIC. BASIC stands for Beginner's All-Purpose Symbolic Instruction Code.

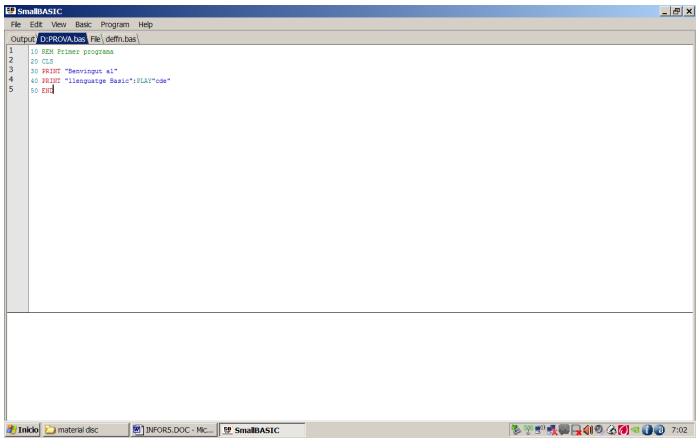
BASIC language is a high level, as well as C and is very suitable for beginners in the world of programming. BASIC is one of the most popular programming languages. It's easy to learn and use, and their orders very similar to English (eg Print, Input) and its power.

SmallBASIC has a simple interface and many mathematical functions and graphics. It is an ideal tool for experimenting with simple algorithms and fun. It has a very simple structure for the programs, their lines can be numbered (10, 20, 30, etc.) are executed in order.

2. Structure of a program in SmallBASIC

```
10 REM Primer Programa
20 CLS
30 PRINT "Benvingut al"
40 PRINT "llenguatge Basic" : PLAY "cde"
50 END
```

We can see the line numbers and the most common commands for separate two commands on one line, use the two points.



Window work SmallBASIC

3. Program Editor

In this section you will learn to work with the publisher of the program:

- Enter the first program
- Save the program file to disk
- Recover files on disk
- Run the program
- Check it properly
- Using Help
- Leave SmallBASIC

Steps to follow:

I. We will create a folder on the USB drive to store programs. For example: Basic.

II. Run the program *My Computer> All Programs> SmallBASIC> SmallBASIC.*

III. The menu *File> New file* and type the following program:

Note: It is very important that you type exactly the program, but the program will not work and get errors

```
10 REM First program "CLS 20

30 PRINT "Welcome to"
40 PRINT "basics": PLAY "CDE" 50 END
```

IV. Stores program previously created the directory, menu *File> Save File As,* how *Exemple.bas.*

V. To skip this step. To retrieve the file from the program menu *File> Open File.*

VI. To run the program press the **F9**

VII. Checking If everything is correct, you get the following result and musical notes, but revised the lines of the program and run it again.

```
Welcome to the basics
```

VIII. Using help: Press **F1.**

IX. To leave SmallBASIC menu File> Exit

Interesting note: In the Edit menu, we have the options: Cut, Copy and Paste, very useful in editing programs, but have previously selected the mouse or text lines.

Keyboard shortcuts for working with the program:

F9 - Run the program
Ctrl + F4 - Close Tabs programs or program information.

Ctrl + B - Stop the program (useful when it is in a loop without leaving the program)

3.1 Programming Course

phases:

1. EX1.bas to EX7.bas.
2. P1.bas to P19.bas.
3. EX8.bas to EX13.bas.
4. EX20.bas to EX26.bas.

4. Major orders SmallBASIC

4.1 Orders screen

REM (remark)

Insert a comment in a program.

CLS (clear screen)

Delete everything on the screen.

COLOR

Change the text color and background of the screen (color code at the end of section 4.5).

COLOR A, B

Change the text color of the screen background color A and B.

PRINT (print)

View numerical data or text on the screen.

example: (EX1.BAS)

```
REM Order 10 PRINT 20 CLS
30 PRINT "HELLO"

40 1.12 COLOR: PRINT 20 + 20
```

Result screen:

```
HELLO 40
```

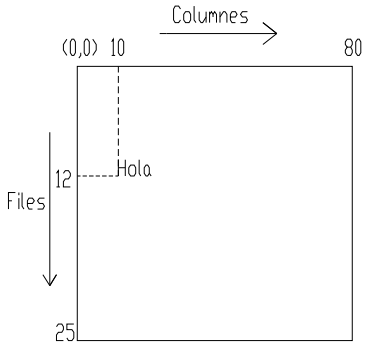
locate

Place the cursor on the screen in a coordinated expressed in rows and columns.

example: (EX2.BAS)

```
10 REM locate command CLS
20
Locate 12.10 30 40
PRINT "HELLO"
```

Result screen:



Placing text and locate PRINT

INPUT

Enter the value of one variable from the keyboard.

example: (EX3.BAS)

```
REM Input Order 10 20 CLS
30 PRINT "Age?"

A 40 INPUT
A 50 PRINT: PRINT "Years"
```

Result screen:

```
Age? ?
15 15
Years
```

(To keep the variable value 15 entered from the keyboard)

STOP

Stop the execution of a program.

END

Ending a program.

PAUSE

Stop the computer.

PAUSE t

Stop the computer during t seconds.

SQR (square root)

Make the square root operation

example: (EX4.BAS)

```
10 20 CLS REM SQR
Order
30 PRINT SQR (25)
```

Result screen:

```
5
```

4.2. Orders Jump

GOTO

Skip the execution of the program line indicated.

example: (EX5.BAS)

```
10 GOTO 20 REM Order
CLS
30 PRINT "HELLO" 40
GOTO 60 50 PRINT
"student" 60 END
```

Result screen:

```
HELLO
```

(Line 50 will not run)

4.3. conditional orders

IF - THEN - ELSE

Forks execution according to the value of an expression. If the value of the expression is true, will run the following command *THEN* and if false run the following *ELSE. ELSE* can be omitted, and simply continue the program in the following order.

example: (EX6.BAS)

```
10 Order REM IF - THEN - ELSE CLS 20

30 PRINT "AGE?" A 40
INPUT
50 IF A> 17 THEN PRINT "adult" ELSE PRINT "MINOR"
```

Result screen:

```
AGE?
11
MINOR
```

4.4. Orders repetition (loops)

FOR - NEXT

Repeat part of a program a certain number of times.

example: (EX7.BAS)

```
Order 10 REM FOR - NEXT 20 CLS

30 FOR I = 1 TO 4 40 PRINT
"GOOD DAY" AND NEXT 50
```

Result screen:

```
GOOD
MORNING
GOOD
MORNING GOOD MORNING GOOD MORNING
```

4.5. orders graphic

Before starting commands graphic graphics grid looks at the end of the chapter, to put points on the screen.

PSET

Draw a dot on the screen.

PSET (*X, Y), color*

Draw a point on the screen coordinates (x, y) and the desired color.

example: (EX8.BAS)

```
REM Command PSET 10 30
40 PSET PSET 320,240,1
320,250,1 320,260,1 60 50
PSET END
```

Result screen:

(Look closely at the three black spots on the screen)

LINE

Draw a line on the screen.

LINE *X1, Y1, X2, Y2, color*

Draw a line on the screen from the initial point (x1, y1) to the end point (x2, y2) and the desired color.

example: (EX9.BAS)

```
Order REM LINE LINE 10 30 40 PAUSE
1 40,40,480,440,14

50 60 END LINE 40,440,480,40,12
```

RECT

Draw a rectangle on the screen.

rect *X1, Y1, X2, Y2, color*

Draw a rectangle on the screen, given by the value of the points of a diagonal from the starting point (x1, y1) to the end point (x2, y2) and the desired color.

example: (EX10.BAS)

```
REM Order RECT RECT 10 30 40
PAUSE 1 40,40,480,440,1

RECT 10,10,100,100,12 50 60 END
```

CIRCLE

Draw a circle on the screen.

***CIRCLE** X, Y, radius, proportion, color*

Draw a circle on the coordinates (x, y), the radius and the desired color. The ratio of a circle equals 1.

example: (EX11.BAS)

```
REM Order 10 20 CLS CIRCLE

PAUSE 1 40 30 CIRCLE 320,200,30,1,14

PAUSE 1 60 50 CIRCLE 320,100,40,1,1

70 80 END CIRCLE 200,100,50,0.5,12
```

PAINT

Give color to a closed area of the screen.

***PAINT** X, Y, color color border area*

Give color to a closed area of the screen coordinates (x, y), and the desired color.

Important: It is mandatory to indicate the color of the border area, if not the same, will color the whole screen.

example: (EX12.BAS)

```
REM command CLS 10 PAINT
20
CIRCLE 100,100,30,1,1 PAUSE 30 40 1
50 60 PAINT 100,100,1 CIRCLE
200,100,40,1,12 70 PAUSE 1


PAINT 80 200 100.12 90 END
```

Note: Orders **RECT** and **CIRCLE,** FILLED option can lead to an end. This option automatically painted inside the rectangle or circle.

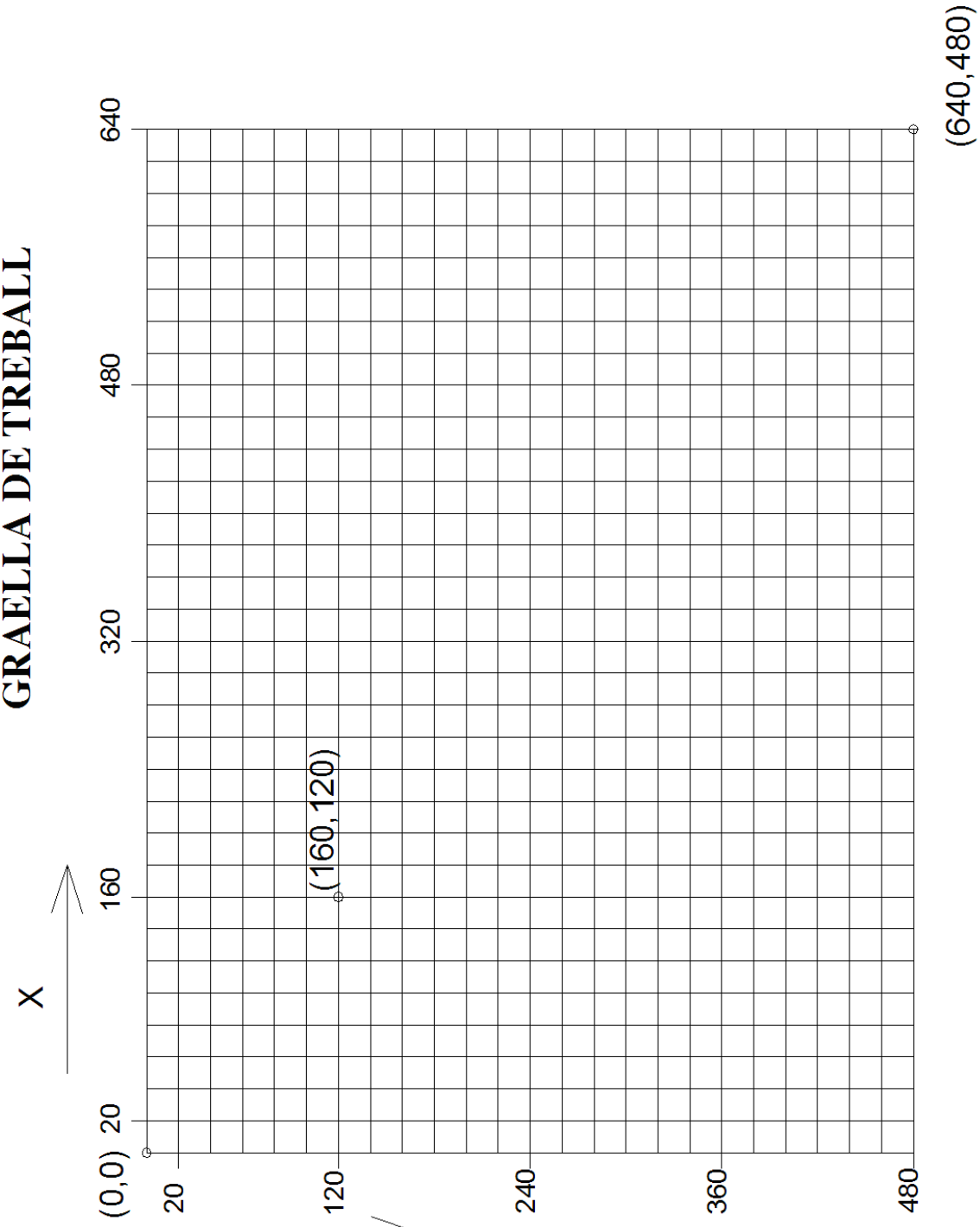
example: (EX13.BAS)

```
10 REM option FILLED CLS 20

100,100,30,1,14 CIRCLE FILLED PAUSE 30 40 1

RECT FILLED 100,100,640,480,1 50 60 END
```

GRAELLA DE TREBALL



CODI DE COLORS
0 negre
1 blau fosc
2 verd fosc
3 cyan fosc
4 roig fosc
5 púrpura fosc
6 marró taronja
7 gris
8 gris fosc
9 blau clar
10 verd clar
11 cyan clar
12 roig clar
13 magenta
14 groc
15 blanc

6. Programming Exercises

Enter these programs on your computer and store it in your USB memory folder. Run each, please tell us the result and performance. Where necessary enter the program.

P1.bas (Operations)

Result screen:

```
10 CLS 20 PRINT 30
PRINT 100-60 3 + 5 * 6 5
40 PRINT 50 PRINT 60
PRINT 120/40 (4 + 5) * 4
```

P2.bas > Write a program that does the following operations:

(230 + 345) / 23

Result screen:

P3.bas (Text and numbers)

Result screen:

```
CLS 10
20 PRINT "I'm a PC" 30 PRINT
"How are you?" 40 PRINT "3 + 5"
```

Because it results in 8?

P4.bas (Locate)

Result screen:

```
CLS 10
Locate 12.40 20: PRINT "A" Locate 12.42
30: PRINT "B" Locate 12.44 40: PRINT
"C"
```

P5.bas> Write a program to make the next screen
row and column indication. (Locate)

30

10

MENU OPTIONS
3 2 1 PLAY exit

P6.bas (Variables, variables are like boxes where you can store values, and then use them)

```
6 A = 20 10 CLS
30 PRINT TO
PRINT 50 7 40 B
= B
```

Result screen:

P7.bas (Sum)

```
CLS 10
20 A = 3: PRINT 30 B =
6: PRINT 40 B C = A +
B
50 PRINT "Sum =" 60
PRINT C
```

Result screen:

P8.bas > Make a program similar to the above to do the sum, product and quotient of two numbers given in variables.

Result screen:

P9.bas (INPUT A and B by entering sum keyboard)

Result screen:

```
CLS 10
20 PRINT "A =" 30
PRINT 40 INPUT A "B ="
50 INPUT C = A + B 60 B
70 PRINT "= SUM" C
PRINT 80
```

P10.bas> Make a program similar to the above to do the sum, product and quotient of two numbers as input variables INPUT.

Result screen:

P11.bas> Make a program to calculate the area of a triangle.

$A = (B \cdot H) / 2$

Result screen:

P12.bas> Make a program that carries out the conversion of euros in pesetas.

1 € = 166.386 pts

Result screen:

P13.bas> Make a program that carries out the conversion of pesetas into euros.

Result screen:

P14.bas> Write a program to calculate the average of three numbers.

Result screen:

tracks:
A INPUT INPUT
INPUT CM B = (A +
B + C) / 3

program:

1st review Note:

4Nota 2nd test: 3rd

test 6Nota: 5Mitja 5

P15.bas (FOR-NEXT loop)

Result screen:

```
CLS 10
20 FOR I = 1 TO 20 30 PRINT
"*"; 40 AND NEXT
```

What happens if the camshafts; which bears the print behind?

P16.bas (FOR-NEXT loop, and locate PRINT)

Result screen:

```
CLS 10
20 FOR I = 1 TO 15 30 Locate I, 15: PRINT
"*" 40 Locate I, 20: print "$" 60 Locate I, 25:
PRINT "@" 70 Locate 20 I: PRINT "%" 80
NEXT I
```

This result explains why women program

P17.bas (Game with GOTO, IF - THEN)

```
CLS 10 X 20
= 6
30 PRINT "************ ************ Guess the number" 40 PRINT "************
                                        (0 to 10)              * * * * * * * * * * * * "
A 50 INPUT
60 IF A = X THEN GOTO 100
70 IF A> X THEN PRINT "---- ---- UNDER" 80 IF A <X THEN
PRINT "---- ---- MAJOR" 90 GOTO 50

100 PRINT "You've almost got $$$$$ $$$$$" PLAY 110
"O2T255CDCDO6DCDC"
```

Result screen:

Write next to each line of the program that performs the function

P18.bas> Make a program that asks you the password (a number) to continue and if the correct sounding music and the text " ***correct password*** "

Result screen:

P19.bas> Program menu: Calculator euros

```
10 REM CALCULATOR CLS 20 EUROS

30 PRINT "EURO CALCULATOR" 40 PRINT "1
EUROS PTS" 50 PRINT "2 PTS Euro" 60 PRINT
"CHOOSE THE OPTION" A 70 INPUT
```

PRINTING THE MENU

```
80 IF A = 1 THEN GOTO 100 90 IF A = 2
THEN GOTO 200 100 EUROS IN REM
CLS 110 PTS
```

TO CHOOSE THE OPTION

```
120 PRINT "EUROS PTS" 130 PRINT
"EURO" INPUT 140 EUROS

PTS = 150 Euro * 166 386 160 PRINT
"PTS =" 170 PRINT END 180 PTS
```

SUBPROGRAMME: PTS IN EUROS

```
REM PTS 200 Euro CLS 210

220 PRINT "PTS Euro" 230 PRINT
"PTS" INPUT 240 PTS

PTS = 250 euros / 166 386 260 PRINT
"EURO =" 270 PRINT END 280 EUROS
```

SUBPROGRAMME: PTS IN EUROS

Result screen:

P20.bas (Order LINE)

Result screen:

```
SCREEN 10 12
20 30 END LINE 5,5,105,105,14
```

P21.bas (Order RECT)

Result screen:

```
CLS 10
RECT 5,5,105,105,14 20 30 END
```

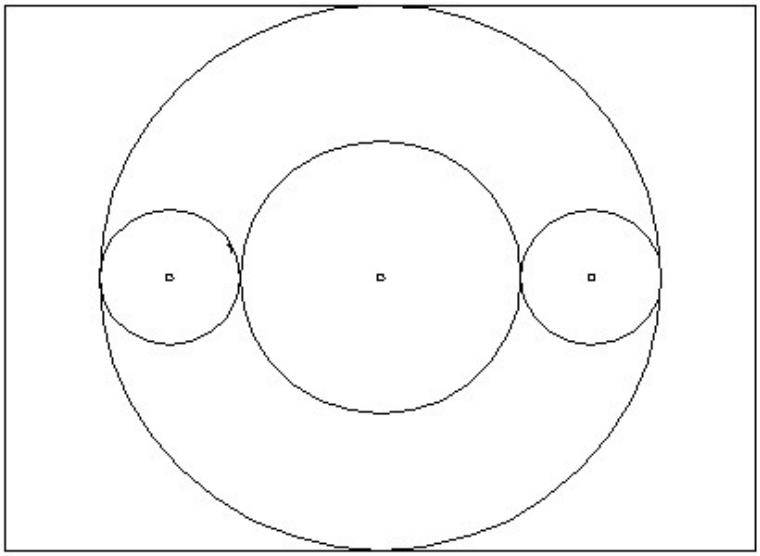
P22.bas (Order RECT, drawing a colored rectangle)

Result screen:

```
CLS 10
RECT FILLED 5,5,105,105,14 20 30 END
```

Explain the difference between the three previous programs

P23.bas> Make a program that does the following figure, each circle a different color (Use the grid on page 10 and make an outline prior to placing coordinates):



P24.bas> Make a program that draws a triangle colored red and yellow I paint inside: