

Lab: Design By Numbers

Labs

Four labs will be held to introduce technical topics. The labs will consist of a brief introduction. Then, several short problems are assigned. Most students should be able to complete the problems in class.

You should create a website and post your solutions there. You will submit this website later in the quarter for review by the instructor.

The first lab concerns DBN (Design By Numbers). The remaining labs concern SVG (Scalable Vector Graphics).

The following is the first lab. Please complete the lab activities at the bottom of this document.

Objective

To complete Assignment #1, you will have to become familiar with Design By Numbers or DBN. DBN is very simple programming language for exploring interactive graphics. With DBN you can create a wide range of visualizations.

The objective of this lab is to:

1. Learn how to edit and run DBN programs
2. Use the following statements in DBN programs:
 - a. Paper
 - b. Pen
 - c. Line
 - d. Variables and calculations
 - e. Set
 - f. Repeat
 - g. Command
 - h. Conditionals

Downloading DBN

DBN is a Java application. You can run it as an Applet here:

<http://dbn.media.mit.edu/dbn/>

Or you can download the Java Application and run it from your machine. We suggest that you use DBN as an application. Download the application from:

<http://dbn.media.mit.edu/download/>

To start the application double click

dbn.bat

This will start the Java virtual machine and run the DBN program.

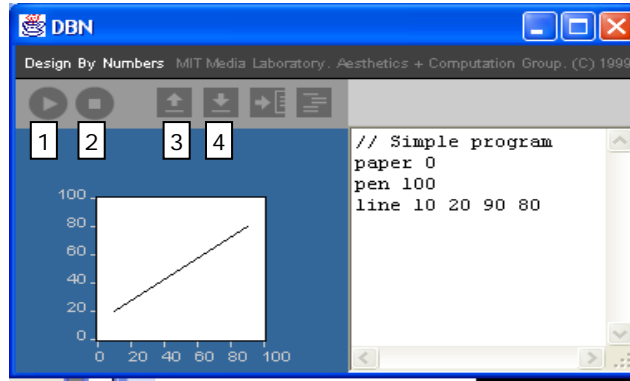
Processing is a more powerful (and complex) language that is actively being used in a number of projects. If you find DBN interesting, you may want to learn more about Processing here:

<http://processing.org/>

Lab: Design By Numbers

Conceptual Model for DBN

The programming environment for DBN is very simple. It consists of a text editor on the right-hand side, where you type programs, and an output display on the left-hand side, which shows the results of the program.



In this case, the program draws a line from the bottom-left to the top-right beginning at the coordinate 10,20 and ending at the coordinate 90, 80. The color of the paper is white (0) and the color of the pen that draws the line is black (100).

The button at (1) is used to run programs and the button at (2) can be used to stop programs that are taking a very long time to finish. The buttons at (3) and (4) are used to open and save programs respectively.

You will find many example DBN programs in the /examples directory and on the web.

For a summary of the DBN language:

<http://dbn.media.mit.edu/info/vocabulary.html>

Hidden features of the language are found here:

<http://dbn.media.mit.edu/info/hidden.html>

The following introduces the basic DBN commands.

Pen and Paper Command

The paper command is used to set the grey scale or opacity of the paper. The value of the paper commands range from 0 for white to 100 for black. In the following program the paper is white and the pen is black; thus, a black line is drawn from the bottom-left to the top-right on white paper:

```
paper 0
pen 100
line 10 20 90 80
```

If the command `paper 50` were used the paper would be 50% grey.

You can also use color paper and pens by using RDB values. Try this program:

```
paper 30 30 100
pen 100 0 0
line 10 20 90 80
```

Lab: Design By Numbers

Line Command

As you might imagine, you use the line command to draw lines. The line command takes four coordinates

```
line x1 Y1 x2 Y2
```

Each point on the paper can be referred to by a pair of numbers. The first number is the x coordinate and the second number is the y coordinate. The origin for the paper is (0,0) in the bottom-left the coordinate for the top-right is (100,100).

Thus, to create a cross over the entire extent of the paper you could submit this program:

```
paper 0  
pen 100  
line 0 0 100 100  
line 0 100 100 0
```

Variables and Calculations

You can use the set command to assign values to variables. Consider the program:

```
paper 0  
pen 100  
set X 10  
set y 20  
line x 0 x 100  
line 0 y 10 y
```

Here the variables x and y are assigned values and these values are used later in the program.

It is also possible to insert calculations into programs as shown here:

```
paper 0  
pen 100  
set X 10  
set y 20  
line (x + 10) 0 x 100  
line 0 y 16 (y + 10)
```

Calculations must always be placed within parentheses. Addition (+), subtraction (-), multiplication (X), and division (/) are allowable operations.

Lab: Design By Numbers

Set Command

You can set the color of any individual pixel on the paper by using the set command. The set command looks like this:

```
set [x y] g
```

Where *x* and *y* are the coordinate for the pixel and *g* is the grey scale value. Thus, this program draws a line from the origin (0,0) towards the top-right. The grey value of the dots varies from black to 50% black:

```
paper 60
set [0 0] 0
set [10 10] 10
set [20 20] 20
set [30 30] 30
set [40 40] 40
set [50 50] 50
```

Repeat Command

The repeat command is used to repeat a command some number of times. It looks like this:

```
repeat var start end
{
    block_of_commands
}
```

where *var* is the name of a variable and *start* and *end* are the starting and ending values of the variable. Block is a set of statements that should be repeated. For example, consider this program which draws 101 vertical lines

```
paper 65
repeat A 0 100 //first sets A to 0, then 1, then 2, etc.
{
    pen A
    line A 30 A 50
}
```

Command Command

The Command *command* is used to create your own commands (sometimes called functions or subroutines). It takes this form:

```
command name p1, p2, ... pN
{
    block_of_commands
}
```

Lab: Design By Numbers

Where *name* is the name of the command and *p1*, *p2* and *pN* are parameters to the command that can then be used in the block_of_commands. For example, this cross command can be used to draw crosses of different sizes in different positions:

```
// Define new function
command cross x y size c
{
  pen c
  line x y x (y+size/2)
  line x y x (y-size/2)
  line x y (x+size/2) y
  line x y (x-size/2) y
}

// Try out function with cross sizes 4, 8, and 20
paper 50
cross 10 10 4 100
cross 80 20 8 100
cross 60 60 20 100
```

Conditional Commands

Same? and NotSame? can be used to conditionally execute blocks of commands. Consider this program:

```
paper 0
pen 100
repeat A 0 100
{
  Same? A 30
  {
    line 100 A A A
  }
  NotSame? A 30
  {
    line A A A 100
  }
}
```

This program will execute the first line statement when A is equal to 30. When A is not equal to 30, the second line statement is executed.

Lab: Design By Numbers

Smaller? and NotSmaller? conditionals that can be used to test a range of values. Consider these two programs:

```
// Program 1
Paper 0
Repeat A 0 100
{
    Pen A
    Smaller? A 50
    {
        Line A 0 A 100
    }
}
// Program 2
Paper 0
Repeat A 0 100
{
    Pen A
    NotSmaller? A 50
    {
        Line A 0 A 100
    }
}
```

Learning more ...

For a summary of the DBN language:

<http://dbn.media.mit.edu/info/vocabulary.html>

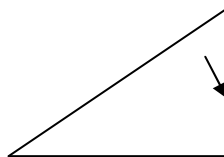
Hidden features of the language are found here:

<http://dbn.media.mit.edu/info/hidden.html>

Lab activities

Create DBN programs that do the following:

1. Draw a triangle using three line statements.
2. Draw another geometric shape.
3. Draw a representational graphic, say a cloud, a flower, or snake.
4. Fill the entire display space with a texture of some kind.
5. Draw a program that shows a line falling down as sketched here:



6. Draw an animation that looks like a bumble bee.