BearPlot documentation and usage

BearPlot v1.4, Mar. 2, 2016

BearPlot is a Python-based program to draw graphics using an input-file-specified graphics primitives. BearPlot is intended to support C++ programs by providing graphics potential as a "second stage of execution" after the C++ program.

This sequence is a kludgy solution to a system problem: that C++ does not have a universal graphics standard; it's difficult to select and install graphics libraries to support the variety of environments needed.

BearPlot contains some error messages when data file lines cannot be interpreted.
BearPlot is licensed by Missouri State University using the MIT license.
Use BearPlot at your own risk.

Tkinter graphic library commands are documented at TKinter documentation at http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html.

Coordinate system for BearPlot graphics

Like many graphics system, BearPlot's Y axis values are REVERSED in comparison to the standard Cartesian coordinate system. The coordinate system origin (0, 0) is at the upper left, and Y values increase going DOWN the graphics window. A line from the origin to the center of the graphics window would be drawn using the coordinates (0, 0) to (WIDTH/2, HEIGHT/2). The conversion from "standard" Cartesian coordinate system value (sx, sy) to graphics coordinate system (gx, gy) is: gx = sx (that is, no difference), gy = (HEIGHT-sy).

Contents of the data file for input to BearPlot graphics

Every data file must contain on the first line a specification of the output graphics window using the form

```
pixelWidth pixelHeight color
```

where `color` is one of the Tkinter known strings (see below for valid color names).
Example data file usage: `700 700 white`

Graphics drawing primitives. The data file may contain whatever of these primitives are needed to display the desired graphic output.

Graphics command. This is the first word or phrase on a line of the data file. *Uppercase or lowercase are allowed.*	**Meaning**	**Data file form and example usage.** *Fields in parentheses are **optional** – when that field does not appear in the data file, some default value is used.*

CIRCLE	Draws a circle, with coordinate bounding box, with top left and diameter given as parameters. Optional fill color, default is transparent (see below for valid color names).	circle topLeftX topLeftY diameter (color) `circle 50 50 20 blue`
COORDINATE PLANE	Draws a basic coordinate plane, spliting the canvas into four equal halves. This command has no parameters other than the name itself.	coordinate_plane
DOTTED LINE	Draws a dotted line from startx starty to endx endy with an optional color as parameter 6. Default color is black. (See below for valid color names).	line_dotted startX startY endX endY (color) `line_dotted 50 50 20 200 blue`
DOTTED RECTANGLE	Draws a rectangle with black dotted border, between two points, top left and bottom right, with optional fill color. Default color is transparent. (See below for valid color names).	rect_dotted topLeftX topLeftY bottomRightX bottomRightY (color) `rect_dotted 50 50 100 100 blue`
LINE	Draws a line from startx and starty to endx and endy with an optional color option. Default color is black. (See below for valid color names).	line startX startY endX endY (optional color, default is black) `line 50 50 20 200 blue`
OVAL	Draws an oval, with coordinates as a bounding box, with top left and bottom right given as parameters. Optional fill color, default is transparent. (See below for valid color names).	oval topLeftX topLeftY bottomLeftX bottomLeftY (color) `oval 50 50 200 100 blue`
POINT	Draws a one pixel point at point x y. With an optional color. Default color is black. (See below for valid color names).	point x y (color) `point 50 50 blue`
POSITIVE COORDINATE PLANE	Draws a basic positive coordinate plane, with the axis 2 pixels from the edge. This command has no parameters other than the name itself.	positive_coordinate_plane
RECTANGLE	Draws a rectangle between two points, top left and bottom right, with optional fill color. Default color is transparent. (See below for valid color names).	rectangle topLeftX topLeftY bottomRightX bottomRightY (color) `rectangle 50 50 100 100 blue`
TEXT	Draws text using black color, centered on point given, numeric	text pointX pointY fontsize word1 word2 word3 . . .

	font size. Typical font size values are integers 10 (small) to 36 (extra-large). The text on the remainder of line will be displayed.	`text 50 50 16 Hello World!`
THICK LINE	Draws a thicker line from startx starty to endx endy with an optional color. Default color is black. (See below for valid color names).	line_solid startX startY endX endY (color) `line_solid 50 50 20 20 blue`
TRIANGLE	Draws segments forming a triangle between three points, optional line color. Default color is black. (See below for valid color names).	triangle point1X point1Y point2X point2Y point3X point3Y (color) `triangle 50 50 20 20 50 20 blue`

Other graphics data file commands.

Graphics command. This is the first word on a line of the data file. *Uppercase or lowercase are allowed.*	**Meaning**	**Example data file usage.** *Fields in parentheses are **optional** – when that field does not appear in the data file, some default value is used.*
COMMENT	Allows comments to be placed in data file. There is no graphical effect resulting from a comment.	comment word1 word2 word3 . . . `comment It certainly is a fine day today!`
PRESSKEY	Pause indefinitely while waiting for user to press ENTER key. *NOTE: Running BearPlot causes system to lose focus into the graphics window. User must first click in DOS or IDLE window to regain focus, then ENTER.*	presskey `presskey`

SLEEP	Pauses execution of graphic display. Optional input in seconds as a floating-point value. Default is 2.	sleep (seconds)
		
		`sleep`
		`sleep 2.5`

Example

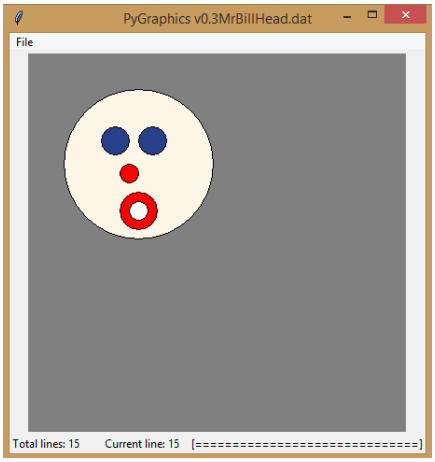
Desired outcome appearance of graphics, the data file used, the result obtained


```
400 400 gray
comment Bill's head
circle 40 40 160 oldlace

comment Bill's eyes
circle 80 80 30 royalblue4
circle 120 80 30
royalblue4

comment Bill's nose
circle 100 120 20 red

comment Bill's mouth
circle 100 150 40 red
circle 110 160 20 white
```



Valid color names

Choose Tkinter color names from color charts found on the web using the search term “Tkinter color name chart”.

Execution of BearPlot software

BearPlot contains some error messages when data file lines cannot be interpreted.

From a DOS shell in a folder containing BearPlot.py and the data file

```
python BearPlot.py datafile.txt
```

From an IDLE window select Run -> Run Module and enter the data filename when prompted